STOCHASTIC DIAGONALIZATION

Raedt, Hans De; Frick, Martin

Published in:
Physics Reports

DOI:
10.1016/0370-1573(93)90015-6

Publication date:
1993

STOCHASTIC DIAGONALIZATION

Hans De RAEDT and Martin FRICK

Institute for Theoretical Physics, University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, The Netherlands

NORTH-HOLLAND

PHYSICS REPORTS (Review Section of Physics Letters) 231, No. 3 (1993) 107–149
North-Holland

PHYSICS REPORTS

Stochastic diagonalization

Hans De Raedt and Martin Frick[1]

Institute for Theoretical Physics, University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, The Netherlands

Received February 1992; editor: M.L. Klein

Contents:

Abstract:

Conventional Quantum Monte Carlo methods, routinely used to compute properties of many-body quantum systems, often suffer from what has been termed 'minus-sign" problems. These problems are shown to result from an elementary property of the Hamiltonian and the use of stochastic (Markovian) simulation methods to compute physical quantities. An importance sampling algorithm is presented to compute the smallest eigenvalue(s) and the corresponding eigenvector(s) of extremely large matrices. It can exploit the sparsity of the solution of the eigenvalue problem. A rigorous proof of the correctness of the algorithm is given. Important aspects of the implementation of the algorithm are discussed at length. Demonstration programs are included. The method is applied to the two-dimensional Hubbard model. The performance of the algorithm is studied in great detail, confirming the expectations based on the theoretical analysis of the algorithm. Results are presented for the smallest eigenvalue and the properties of the corresponding eigenvector of matrices of order up to $10^{35} \times 10^{35}$.

[1] Present address: Thinking Machines Corporation, 245 First Street, Cambridge MA 02142-1264, USA.

1. Introduction

The most direct approach to investigate quantum many-body systems is to solve the eigenvalue problem of the Hamiltonian $\mathscr{H}$ by means of standard diagonalization techniques [1]. The limitations of this approach become obvious by considering a typical many-body system of $N_\uparrow (N_\downarrow)$ electrons with spin up (down) distributed over L orbitals, often corresponding to L lattice sites. In this case the dimension of the Hilbert space is given by $N = \binom{L}{N_\uparrow} \times \binom{L}{N_\downarrow}$, a rapidly increasing function of L if $1 \ll N_\uparrow \ll L$ or $1 \ll N_\downarrow \ll L$. The dimension of the matrix H representing the Hamiltonian is $N \times N$. For instance taking $L = 16$ and $N_\uparrow = N_\downarrow = 5$ yields $N \approx 19 \times 10^6$ whereas for $L = 64$ and $N_\uparrow = N_\downarrow = 13$, $N \approx 17 \times 10^{25}$. Storage and CPU-time requirements of standard diagonalization algorithms are proportional to N^2 and N^3, respectively, effectively limiting the applicability of this straightforward approach to systems represented by rather small matrices H.

For many but not all quantum many-body problems of interest it is often sufficient to calculate the ground-state properties. Then the Lanczos [1, 2, 3] (inverse) power [1, 2] or (generalized) Davidson [4, 5] method may be used to compute the ground state, i.e., the eigenvector corresponding to the lowest eigenvalue of H. Compared to standard techniques, an important advantage of this set of methods is that the storage needed is proportional to N instead of N^2 and the CPU-time required to iterate to the ground state scales with N^2 instead of N^3. The systems that can be studied by means of these methods are significantly larger than those amenable to brute force diagonalization.

As the many-particle system becomes larger, N grows (exponentially) fast. Going beyond the limit (to be denoted by N_L set by current computer technology and the algorithms mentioned above, requires a conceptual change of strategy. From a computational point of view, the problem may be considered as unsolvable if one has to perform an accurate calculation of each of the $N \gg N_L$ elements of the vector, representing the ground state. To make progress, it is necessary to make the hypothesis that the problem is solvable. If out of the N possible configurations (states) knowledge of only a small fraction $N_R (N_R \leq N_L \ll N)$ of relevant states suffices to compute the ground-state properties to the desired accuracy, the computational problem is to find these N_R states and to obtain information to decide if the fundamental hypothesis is correct or not.

Searching the very large set of N states for the N_R relevant states may be viewed as a problem of importance sampling. Unfortunately, the probability of a state to occur, i.e., its contribution to the ground state, is unknown. In classical equilibrium statistical mechanics one faces a similar problem. The probability for a configuration is $p_j \equiv e^{-\beta E_j}/\Sigma_j e^{-\beta E_j}$ where E_j is the energy corresponding to the configuration j. The partition function $Z \equiv \Sigma_i e^{-\beta E_i}$ is, in general, unknown and hence so is p_j. Any Markov process which has $\{p_j\}$ as its limit distribution can be used to generate the "important" configurations, i.e., those that give the largest contributions to Z. The Metropolis Monte Carlo (MMC) method [6, 7, 8] is the most widely used algorithm implementing this idea but other simulation techniques such as Molecular Dynamics (MD) or Langevin Dynamics (LD) can be used as well. The MMC method uses the ratio p_i/p_j to determine the transition probability for the underlying stochastic process. Crucial thereby is that in forming the ratio, the unknown partition function drops out.

The apparent similarity between quantum and classical problems can be exploited by reformulating the calculation of the lowest eigenvalue or thermal expectation values of a quantum system as a Markov process on the space of states. This is usually done by invoking the path integral or Trotter–Suzuki representation [9]. These simulation techniques can be used to compute either zero temperature or nonzero temperature properties. The Markov process will properly sample the important contributions to the ground state provided the elements of the stochastic matrix, defining the Markov process, correspond to the matrix elements of a judiciously chosen function $f(H)$ of the Hamiltonian H. However, for many problems of interest this correspondence seems extremely hard to find. The fundamental reason is that the elements of the stochastic matrix, being probabilities, have to be positive whereas the matrix elements of $f(H)$ calculated using a particular representation of the states, may differ in sign. As explained in more detail below, this fundamental difficulty leads to the so-called minus-sign problem. It results from the choice of the representation used to calculate the matrix elements of $f(H)$, in combination with the desire to use a Markov chain as a vehicle to search for the important states. Note that this rather general discussion suggests that the minus-sign problem should also manifest itself in cases where there are no fermionic degrees of freedom. Indeed, there are ample examples supporting the point of view that the minus-sign problem is of more general nature.

From the preceeding discussion it is clear that in order to circumvent the minus-sign problem one may have to abandon the idea of using a Markov process or a MD technique to search for important states. In this paper a detailed exposition is given of a method that uses a process, defined in terms of orthogonal instead of stochastic matrices, to collect the important contributions to the ground state. It is free of minus-sign problems and allows the study of systems larger than those accessible by Lanczos or (inverse) power methods. It exploits the "sparseness" of the solution instead of sparseness of the matrix.

The article is organized as follows. In chapter 2 a general discussion is given of the origin of minus-sign problems encountered in Quantum Monte Carlo (QMC) work. The reader who is not familiar with QMC methods may skip this section. The theoretical description of our method, which we call stochastic diagonalization (SD), is presented in chapter 3. Chapter 4 discusses the implementation of the algorithm and also contains some demonstration programs. Chapter 5 presents results for the two-dimensional (2D) Hubbard model. The performance of the algorithm is compared to the theoretical predictions, given in chapter 4. The algorithm is used to compute the ground-state energy and static correlation functions of the 2D Hubbard model. These results are compared to data obtained by other means. A summary and conclusions are given in chapter 6. Brief reports on parts of the material presented in this article have been published elsewhere [10].

2. Minus-sign problem in Quantum Monte Carlo methods

The origin of the minus-sign problem can be identified by considering one of the QMC methods (excluding variational QMC). The concepts to be used are sufficiently general to allow the reader to apply the same reasoning to his favourite QMC technique. Consider the Projector Quantum Monte Carlo (PQMC) scheme. The ground state $|\Phi_0\rangle$ of the system, described by the Hamiltonian H, is given by $|\Phi_0\rangle = \lim_{\beta\to\infty} |\phi(\beta)\rangle$ whereby

$$|\phi(\beta)\rangle = \mathrm{e}^{-\beta H}|\phi_0\rangle \langle \mathrm{e}^{-\beta H}\phi_0 | \mathrm{e}^{-\beta H}\phi_0\rangle^{-1/2}, \tag{2.1}$$

and ϕ_0 is some state for which $\langle\phi_0|\Phi_0\rangle \neq 0$. Furthermore it has been assumed that the ground state is nondegenerate. Straightforward application of (2.1) requires the same amount of storage as the Lanczos, power or Davidson technique. In addition it would require the invention of a technique to evaluate $e^{-\beta H}$ for very large matrices H and large β.

The storage problem and the difficulties related to the calculation of $e^{-\beta H}$ may be circumvented by considering matrix elements instead of vectors. Ground state expectation values generally are of the form

$$\langle A\rangle \equiv \langle\Phi_0|A|\Phi_0\rangle = \lim_{\beta\to\infty}\langle A\rangle_\beta\,, \quad \langle A\rangle_\beta = \langle\phi(\beta)|A|\phi(\beta)\rangle = \frac{\langle e^{-\beta H}\phi_0|A|e^{-\beta H}\phi_0\rangle}{\langle e^{-\beta H}\phi_0|e^{-\beta H}\phi_0\rangle}\,. \tag{2.2a, b}$$

The ground state energy, i.e. the smallest eigenvalue of H, is given by $\langle H\rangle$. Interpreting $|\phi(\beta)\rangle$ as a trial wave function, it is clear that the projector approach will yield an upper bound to the ground-state energy (i.e. $\langle H\rangle \leq \langle H\rangle_\beta$) if the numerator and denominator in (2.2b) can be calculated exactly. In practice, both quantities are usually obtained from simulation and are subject to statistical errors. Then the nice feature of having an upper bound to the ground-state energy may be lost.

Inserting resolutions of the identity, i.e., $\Sigma_{\{\psi_i\}}|\psi_i\rangle\langle\psi_i| = 1$ and $\Sigma_{\{\psi_i'\}}|\psi_i'\rangle\langle\psi_i'| = 1$, (2.2b) can be written as $\langle A\rangle_\beta = \lim_{m\to\infty}\langle A\rangle_{\beta,m}$, where

$$\langle A\rangle_{\beta,m} = \sum_{\{\psi_i\}}\sum_{\{\psi_i'\}}\frac{\langle\psi_m|A|\psi_m'\rangle}{\langle\psi_m|\psi_m'\rangle}\rho(\tau, m, \phi_0, \{\psi_i, \psi_i'\})\Big/\sum_{\{\psi_i\}}\sum_{\{\psi_i'\}}\rho(\tau, m, \phi_0, \{\psi_i, \psi_i'\})\,, \tag{2.3a}$$

$$\rho = \rho(\tau, m, \phi_0, \{\psi_i, \psi_i'\}) = \langle\phi_0|e^{-\tau H}|\psi_1\rangle\cdots\langle\psi_{m-1}|e^{-\tau H}|\psi_m\rangle \times\langle\psi_m|\psi_m'\rangle\langle\psi_m'|e^{-\tau H}|\psi_{m-1}'\rangle\cdots\langle\psi_1'|e^{-\tau H}|\phi_0\rangle\,. \tag{2.3b}$$

In (2.3) the calculation of matrix elements of $e^{-\beta H}$ has been replaced by sums over all possible states $S = \{\psi_i, \psi_i'\}$ of products of matrix elements of $e^{-\tau H}$. In practice it is convenient to choose the sets of states such that $\langle\psi_m|\psi_m'\rangle = \delta_{\psi_m,\psi_m'}$. There are several techniques to compute the matrix elements of $e^{-\tau H}$ to high accuracy if τ is "small enough". In general the memory requirements for evaluating (2.3) are modest.

PQMC uses $e^{-\beta H}$ as a vehicle to filter out the ground state from a state $|\phi_0\rangle$, but any other filter $f(H)$ could be used as well. QMC methods compute $f(H)$ by approximating it by means of a product formula, i.e., $f(H) \approx [g(H/m)]^m$. The function g is chosen such that matrix elements of $g(H/m)$ are easy to calculate. Most QMC techniques employ $f(H) = e^{-\tau H}$ as a filter. Then the standard approach to approximate $e^{-\tau H}$ is to use a Trotter–Suzuki formula [9, 11, 12]. For instance, if $H = H_0 + H_1$, $e^{-\tau H} \approx e^{-\tau H_0}e^{-\tau H_1}$ is the simplest approximation. The choice of the product formula may be crucial for the application of a QMC method but it is of little relevance for the discussion of the minus-sign problem.

In most applications the number of contributions to the sums in (2.3a) is extremely large ($\propto N^{2m}$) so that brute force summation of all terms is not possible. The alternative is to limit the sums to the dominant contributions. If $\rho > 0$ for all $S \equiv \{\psi_i, \psi_i'\}$, (2.3a) can be written as

$$\langle A\rangle_{\beta,m} = \sum_S A_{\beta,m}(S)\,e^{-\beta E(S)}\Big/\sum_S e^{-\beta E(S)}\,, \quad \beta E(S) = -\ln\rho\,. \tag{2.4}$$

Equation (2.4) makes explicit the formal equivalence between the calculation of the ground-state expectation value of an observable A and the computation of an expectation value within the framework of classical statistical mechanics. The ratio (2.4) of the two sums over all possible configurations S can be estimated by means of MMC or MD simulation.

The analogy with classical statistical mechanics breaks down if ρ can take negative values. The fundamental difficulty, i.e., the minus-sign problem, is that quantities of interest are sums of a positive and a negative contribution which, unfortunately, nearly cancel each other. Extremely good statistics and accuracy may be required to obtain meaningful results.

It is of interest to consider the question under which conditions QMC techniques will suffer from the minus-sign problem. Examples show that it may be difficult to specify sufficient conditions but a necessary condition is much easier to find. According to (2.3b), $\rho < 0$ requires that at least one of the matrix elements of the type $\langle\phi|e^{-\tau H}|\psi\rangle$ is negative. The following theorem [13] is useful.

Theorem 1. The necessary and sufficient condition for $\langle\phi|e^{-\tau H}|\psi\rangle$ to be positive for all $\tau > 0$ is $\langle\phi|H|\psi\rangle \leq 0$ for all $\phi \neq \psi$.

A proof of this theorem is given in appendix A. Note that the necessary condition explicitly depends on the choice of the representation of the states of the system. Theorem 1 predicts when a particular factor in the expression (2.3b) for ρ may become negative, but it may still happen that the product of all factors is positive, because the total number of negative factors is even.

Stochastic implementations of the inverse power method, e.g. the Green Function Monte Carlo (GFMC) technique [14], possibly use matrix elements of the inverse of the Hamilonian. The following theorem [13] specifies the necessary condition for these algorithms to be free of minus-sign problems.

Theorem 2. If ω is taken such that $\omega + H$ is a positive-definite matrix with the property that $\langle\phi|H|\psi\rangle \leq 0$ for all $\psi \neq \phi$, then $(\omega + H)^{-1}$ has all positive elements.

The proof can be found in appendix A. If $(\omega + H)^{-1}$ has all positive elements it can, in principle, be used to generate a Markov process [15]. Clearly, the conditions on H for theorem 1 or theorem 2 to hold are essentially the same.

The necessary condition for not having minus-sign problems seems rather restrictive. There are a number of examples where, at first glance, the necessary condition is not fulfilled but where there are no minus-sign problems. This is, in all cases that we know of, due to the presence of symmetries that allow us to reverse the sign of the nondiagonal elements of H by changing the representation of the states.

The above discussion of the minus-sign problem focused on zero-temperature QMC methods. In fact it applies equally well to finite-temperature QMC schemes. QMC techniques based on expressions of the partition function obtained by analytical summation (or integration) over part of the degrees of freedom are more difficult to analyse. In most cases, notably fermion systems, function ρ then contains one or more determinants of nonsymmetric real matrices. The eigenvalues of these matrices are not necessary real, possibly leading to the minus-sign problem.

The conclusion is that the minus-sign problem results from the combination of the presence of positive nondiagonal matrix elements of H, a property that strongly depends on the choice of representation of the states, and the necessity to use important sampling techniques based on Markov processes to compute estimators of physical quantities.

3. Theory of stochastic diagonalization

This chapter is concerned with the theory of the stochastic diagonalization technique. The basic ingredients of the method are introduced in three steps (sections 3.2–3.4), each one being an essential part in the construction of a rigorous proof of the correctness of the algorithm. The material is matrix theory rather than a description of an implementation of the algorithm. The latter can be found in chapter 4.

Before beginning let us first introduce some items of notation. The projection of a real and symmetric matrix H on the subspace spanned by the $n \leq N$ orthonormal states (vectors) $S^{(n)} \equiv \{\phi_1, \ldots, \phi_n\}$ will be denoted by $H^{(n)}$, matrix elements by $H_{ij}^{(n)} = \langle \phi_i | H | \phi_j \rangle$ and the eigenvalues $E_i^{(n)}$ of $H^{(n)}$ are assumed to be ordered such that $E_1^{(n)} \leq E_2^{(n)} \leq \cdots \leq E_n^{(n)}$. Eventually the superscript n will keep track of the number of important states and therefore also of the size of the matrix $H^{(n)}$. Evidently $H^{(N)} = H$ and $E_i^{(N)} = E_i$. We will use H and H_{ij} to denote the full matrix and the matrix elements, respectively. Without loosing generality we may assume that in each row (or column) $i = 1, \ldots, N$ there is at least one nondiagonal matrix element (H_{ij}) that differs from zero, i.e.,

$$\sum_{j \neq i} |H_{ij}| > 0 , \quad 1 \leq i \leq N . \tag{3.1}$$

Otherwise the matrix (block) decomposes into smaller matrices. Then the determination of the smallest eigenvalue requires the calculation of the smallest eigenvalue of each block. The transpose of a matrix A is denoted by A^{T}. The norm of a vector $x = (x_1, \ldots, x_n)$ will be denoted by $\| x \| = (\Sigma_{i=1}^{n} x_i^2)^{1/2}$ and $\| A \|$ stands for the spectral norm, i.e., the square root of the largest eigenvalue of the matrix $A^{\mathrm{T}} A$ [1].

A plane rotation involving states ϕ_i and ϕ_j ($1 \leq i < n$, $i < j \leq n$) is represented by an $n \times n$ orthogonal matrix $U^{(n,k)}$ which, in block matrix form, can be written as

$$U^{(n,k)} = U^{(n,k)}(i_{n,k}, j_{n,k}, c_{n,k}, s_{n,k}) = \begin{array}{c} \begin{array}{ccccccc} & 1 & & i_{n,k} & & j_{n,k} & n \end{array} \\ \left(\begin{array}{ccccccc} 1 & & & & & & \\ & \cdots & & & & & \\ & & & c_{n,k} & \cdots & s_{n,k} & \\ & & & \vdots & 1 & \vdots & \\ & & & -s_{n,k} & \cdots & c_{n,k} & \\ & & & & & & \cdots \\ & & & & & & 1 \end{array} \right) \end{array} . \tag{3.2}$$

In (3.2) all diagonal elements are unity except for the two elements $c_{n,k}$ in columns $i_{n,k}$ and $j_{n,k}$. All nondiagonal elements are zero except the two elements $-s_{n,k}$ and $s_{n,k}$. The subscript k will be used as a running index of the plane rotations for fixed dimension n. This admittedly complicated notation is necessary to avoid ambiguities in the interpretation of the symbols. The product of a sequence of plane rotations will be denoted by

$$\mathcal{U}^{(n,m)} = U^{(n,1)} \cdots U^{(n,m)} . \tag{3.3}$$

We adopt the convention that the order in which plane rotations are applied corresponds to the value of k, i.e., first $U^{(n,1)}$, then $U^{(n,2)}$ and so on. The transformed matrix is given by

$$H^{(n,m)} = [\mathscr{U}^{(n,m)}]^{\mathrm{T}} H^{(n)} \mathscr{U}^{(n,m)} . \tag{3.4}$$

Note that the label n only determines the dimension of the matrices and that it puts no restriction on m. Plane rotations will be determined by the following elementary result.

Lemma 1. The eigenvalues $\lambda_1 \leq \lambda_2$ of a real and symmetric matrix $A = \begin{pmatrix} x & y \\ y & z \end{pmatrix}$ where $x \leq z$ and $y \neq 0$ satisfy $\lambda_1 < x \leq z < \lambda_2$.

The eigenvalues of A are

$$\lambda_1 = x - ty < x , \qquad \lambda_2 = z + ty > z , \tag{3.5a, b}$$

and the orthogonal matrix U given by

$$U = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \qquad c = \frac{1}{\sqrt{1+t^2}}, \quad s = \frac{t}{\sqrt{1+t^2}},$$

$$t = \frac{2y}{z - x + \sqrt{(z-x)^2 + 4y^2}}, \quad |t| \leq 1 , \tag{3.5c, d, e}$$

diagonalizes the matrix A, i.e.

$$U^{\mathrm{T}} A U = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}. \tag{3.5f}$$

The quantities defined in (3.5) have been written such that under all circumstances their calculation is numerically stable. The strict inequality $\lambda_1 < x$ will be essential for the construction of the importance sampling algorithm.

To compute the ground state, it is sufficient to construct an orthogonal transformation that reduces an $N \times N$ real symmetric matrix H to the form

$$U^{\mathrm{T}} A U = \begin{pmatrix} E_1 & 0^{\mathrm{T}} \\ 0 & \tilde{H} \end{pmatrix}, \tag{3.6}$$

where E_1 is the smallest eigenvalue of H, $0^{\mathrm{T}} = (0, \ldots, 0)$ is a null-vector and $\tilde{H}$ is some $(N-1) \times (N-1)$ matrix. Here and in the following a tilde on a symbol indicates that the explicit knowledge of the object is not required. The method described below is based on the construction of an orthogonal transformation (i.e. a sequence of plane rotations) that bring $H^{(n)}$ to the form

$$U^{\mathrm{T}} H^{(n)} U = \begin{pmatrix} E_1^{(n)} & 0^{\mathrm{T}} \\ 0 & \tilde{H} \end{pmatrix}. \tag{3.7}$$

Furthermore these transformations can be used as a vehicle to decide which states are "important" and which are not. This then leads to a well-defined procedure to enlarge the set of $S^{(n)}$ of "important states". The combination of these two procedures transforms the $N \times N$ matrix H to the form (3.6).

3.1. Classical Jacobi method

Disregarding all aspects related to the selection of the important states, the stochastic diagonalization (SD) method presented may be viewed as a variant of the Jacobi technique specifically designed to compute the smallest eigenvalue *). In the classical Jacobi method [1] the matrix H

$$H = H^{(N)} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1N} \\ H_{12} & H_{22} & \cdots & H_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ H_{1N} & H_{2N} & \cdots & H_{NN} \end{pmatrix} \tag{3.8}$$

is transformed to diagonal form by a sequence of plane rotations, i.e.

$$\lim_{m \to \infty} H^{(N,m)} = \begin{pmatrix} E_{i_1} & 0 & \cdots & 0 \\ 0 & E_{i_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & E_{i_N} \end{pmatrix}, \tag{3.9}$$

where $\{i_p; p = 1, \ldots, N\}$ is some permutation of the N indices. In the original Jacobi approach the pairs $(i_{n,k}, j_{n,k})$ are chosen such that the off-diagonal element of maximum modulus is reduced to zero. Each plane rotation modified two rows and two columns of the matrix. For reasons of computational efficiency it is expedient to annihilate elements in strict order. Rotations are carried out only if the modulus of the off-diagonal element exceeds some threshold value. This algorithm is called the cyclic Jacobi method.

The convergence of the classical Jacobi method follows from the observation that with each plane rotation, the sum of the squares of the diagonal elements increases monotonically and, since $\Sigma_{i,j=1}^{N} H_{ij}^2$ is constant, the sum of the squares of the nondiagonal elements converges to zero [1].

3.2. Modified Jacobi method

As already mentioned, a strategy to compute the ground state would be to transform the matrix H in

$$U^{\mathrm{T}} H U = \begin{pmatrix} E_1 & 0^{\mathrm{T}} \\ 0 & \tilde{H} \end{pmatrix}. \tag{3.10}$$

We now examine a modification of the cyclic Jacobi method that might accomplish this. To sketch the idea let us assume for a moment that $H_{11} \leq H_{jj}, 2 \leq j \leq N$. We will remove this restriction

*) A minor modification is required to compute the largest eigenvalue, see lemma 1.

later. If, instead of considering all pairs (i,j), the plane rotations involve pairs $(1,2),\ldots,(1,N)$ only, then

$$\hat{H}^{(N)} \equiv \lim_{m\to\infty} H^{(N,m)} = \begin{pmatrix} E_i & 0^{\mathrm{T}} \\ 0 & \tilde{H} \end{pmatrix}, \tag{3.11}$$

where $E_i = E_i^{(N)}$ is one of the eigenvalues. The proof of (3.11) is straightforward. According to lemma 1, application of a plane rotation involving a pair $(1,j)$ strictly reduces $H_{11}^{(N,m)}$, i.e.

$$H_{11}^{(N,m+1)} < H_{11}^{(N,m)}, \quad m > 0, \tag{3.12a}$$

and since

$$E_1^{(N)} \leq H_{11}^{(N,m)}, \tag{3.12b}$$

the sequence $\{H_{11}^{(N,m)}\}$ is monotonically decreasing and bounded from below. Thus $\lim_{m\to\infty} H_{11}^{(N,m)} = \hat{E}$ exists. Furthermore $\lim_{m\to\infty} H_{1j}^{(N,m)} = 0$ for all $j\in\{2,\ldots,N\}$. To prove this statement assume the contrary, i.e., $\lim_{m\to\infty} H_{1j}^{(N,m)} \neq 0$ for at least one $j\in\{2,\ldots,N\}$. Then, according to lemma 1, a plane rotation involving the pair $(1,j)$ would reduce the (1, 1) element, in contradiction with the assumption that the monotonically decreasing sequence $\{H_{11}^{(N,k)}\}$ converges to $\hat{E}$. Moreover, since $\lim_{m\to\infty} H_{1j}^{(N,m)} = 0$ for all $j\in\{2,\ldots,N\}$, $\hat{E}$ is an eigenvalue. Hence $\hat{E} = E_i^{(N)}$ for some $i\in\{1,\ldots,N\}$. This completes the proof that this variant of the Jacobi method isolates an eigenvalue.

The eigenvector corresponding to $E_i^{(N)}$ is given by $\Phi_i^{(N)}$ with

$$\Phi_i^{(n)} \equiv \lim_{m\to\infty} \sum_{j=1}^{n} \mathscr{U}_{ji}^{(n,m)} \phi_j, \quad 1 \leq n \leq N, \tag{3.13}$$

i.e., the ith column vector of $\mathscr{U}^{(n,m)}$ in the basis $\{\phi_1,\ldots,\phi_n\}$.

3.3. *Matrix inflation*

We have seen that the modified Jacobi method isolates an eigenvalue and yields the corresponding eigenvector. In order to obtain the smallest eigenvalue we must use an additional device. The key idea is to combine the modified Jacobi method with the process of increasing the size of the matrix. The latter will also enable us to determine which states are important and which are not.

The theoretical justification of the method is by induction. Consider the submatrix

$$H^{(n)} = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1n} \\ H_{12} & H_{22} & \cdots & H_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ H_{1n} & H_{2n} & \cdots & H_{nn} \end{pmatrix}, \tag{3.14}$$

and assume that application of the modified Jacobi scheme reduces $H^{(n)}$ to the form

$$\hat{H}^{(n)} \equiv \lim_{m\to\infty} H^{(n,m)} = \begin{pmatrix} E_1^{(n)} & 0^{\mathrm{T}} \\ 0 & \tilde{H} \end{pmatrix}, \tag{3.15}$$

where $E_1^{(n)}$ is the smallest eigenvalue of $H^{(n)}$. This assumption is trivially satisfied for $n = 1$. We now inflate the matrix $H^{(n)}$ by adding the $(n+1)$th row and column.

Then, apply to $H^{(n+1)}$ the sequence of plane rotations that transforms $H^{(n)}$ to the form (3.15) and obtain

$$\hat{H}^{(n+1)} = \begin{pmatrix} \hat{\mathscr{U}}^{(n)} & 0 \\ 0 & 1 \end{pmatrix}^{\mathrm{T}} H^{(n+1)} \begin{pmatrix} \hat{\mathscr{U}}^{(n)} & 0 \\ 0 & 1 \end{pmatrix} \tag{3.16a}$$

$$= \begin{pmatrix} E_1^{(n)} & 0 & \cdots & 0 & \alpha_1^{(n+1)} \\ 0 & \tilde{H}_{22} & \cdots & \tilde{H}_{2n} & \alpha_2^{(n+1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \tilde{H}_{2n} & \cdots & \tilde{H}_{nn} & \alpha_n^{(n+1)} \\ \alpha_1^{(n+1)} & \alpha_2^{(n+1)} & \cdots & \alpha_n^{(n+1)} & H_{n+1\,n+1} \end{pmatrix}, \tag{3.16b}$$

$$\hat{\mathscr{U}}^{(n)} = \lim_{m\to\infty} \mathscr{V}^{(n,m)}, \qquad \alpha_j^{(n+1)} = \lim_{m\to\infty} \alpha_j^{(n+1,m)}, \quad j = 1, \ldots, n, \tag{3.17a, b}$$

$$\alpha_j^{(n+1,m)} = \left[\begin{pmatrix} \mathscr{V}^{(n,m)} & 0 \\ 0 & 1 \end{pmatrix}^{\mathrm{T}} H^{(n+1)} \begin{pmatrix} \mathscr{V}^{(n,m)} & 0 \\ 0 & 1 \end{pmatrix} \right]_{j\,n+1}, \tag{3.17c}$$

$$= \left[\begin{pmatrix} \mathscr{V}^{(n,m)} & 0 \\ 0 & 1 \end{pmatrix}^{\mathrm{T}} H^{(n+1)} \right]_{j\,n+1} = \left[H^{(n+1)} \begin{pmatrix} \mathscr{V}^{(n,m)} & 0 \\ 0 & 1 \end{pmatrix} \right]_{n+1\,j}, \tag{3.17d, e}$$

$$\mathscr{V}^{(n,m)} \equiv \begin{pmatrix} \hat{\mathscr{U}}^{(n-1)} & 0 \\ 0 & 1 \end{pmatrix} \mathscr{U}^{(n,m)}, \qquad \hat{\mathscr{U}}^{(1)} = 1. \tag{3.17f}$$

Here use has been made of the symmetry of H and the fact the plane rotations in (3.17) do not affect the matrix elements in column $n+1$.

Let us now assume that

$$\alpha_1^{(n+1)} \neq 0. \tag{3.18}$$

We will discuss the case $\alpha_1^{(n+1)} = 0$ in more detail below. According to lemma 1, a single plane rotation involving the pair $(1, n+1)$ will lead to a reduction of the (1, 1) element of $H^{(n+1)}$ provided

$$E_1^{(n)} \leq H_{n+1\,n+1}, \tag{3.19}$$

a restriction to be removed later. With $x = E_1^{(n)}$, $y = \alpha_1^{(n+1)}$, and $z = H_{n+1\,n+1}$, (3.5) yields

$$H^{(n+1,1)} = [U^{(n+1,1)}(1, n+1, c_{n+1,1}, s_{n+1,1})]^{\mathrm{T}} \hat{H}^{(n+1)} \times U^{(n+1,1)}(1, n+1, c_{n+1,1}, s_{n+1,1}) , \tag{3.20a}$$

$$= \begin{pmatrix} \lambda_1 & \beta^{\mathrm{T}} & 0 \\ \beta & \tilde{H} & \tilde{\gamma} \\ 0 & \tilde{\gamma}^{\mathrm{T}} & \lambda_2 \end{pmatrix}, \quad \beta^{\mathrm{T}} = -s_{n+1,1}(\alpha_2^{(n+1)}, \ldots, \alpha_n^{(n+1)}) , \tag{3.20b}$$

$$\lambda_1 < E_1^{(n)} . \tag{3.21}$$

As shown in appendix A, application of the separation theorem [1] gives

$$E_1^{(n+1)} \leq E_1^{(n)} . \tag{3.22}$$

and hence

$$E_1^{(n+1)} \leq \lambda_1 < E_1^{(n)} . \tag{3.23}$$

If $\beta = 0$ we have $E_1^{(n+1)} = \lambda_1$. In general $\beta \neq 0$ but we know [see the discussion following eq. (3.10)] that in the modified Jacobi method the (1, 1) element monotonically decreases and converges to an eigenvalue. According to inequality (3.23), application of the modified Jacobi strategy to the matrix (3.20) will yield the smallest eigenvalue of $H^{(n+1)}$, i.e., $\lim_{m\to\infty} H_{11}^{(n+1,m)} = E_1^{(n+1)}$. Then returning to (3.16) with n replaced by $n+1$, the whole procedure can be repeated. This completes the inductive proof that the method will isolate the smallest eigenvalue of H.

In theory, the calculation starts by diagonalizing the 2×2 matrix. Then one row and column is added to the matrix and the modified Jacobi method is employed to compute the smallest eigenvalue of the 3×3 matrix. This step is repeated, yielding the smallest eigenvalue of a 4×4 matrix, 5×5 matrix, and so on. Evidently, to be useful, an implementation of this scheme requires some additional modifications. These will be discussed in section 3.4.

We now return to the assumptions made in the course of devising the method. Restriction (3.19) (which includes the condition $H_{11} \leq H_{jj}$) is trivially removed. If this condition is not satisfied, application of the permutation

$$P = \begin{pmatrix} 1 & \cdots & n+1 & \cdots & N \\ n+1 & \cdots & 1 & \cdots & N \end{pmatrix}, \tag{3.24}$$

will bring the matrix in the desired form, without loosing numerical stability. In practice, performing this operation is trivial.

At each inflation step ($n \to n+1$) we may have $\alpha_1^{(n+1)} = 0$. Then the arguments that were used to prove convergence to the smallest eigenvalue cannot be used because inequality (3.21) does not hold. If the matrix is block-diagonal, i.e. $\alpha_j^{(n+1)} = 0$ for all $j \in \{1, \ldots, n\}$, it is clear that we have to compute the lowest eigenvalue of each block. However this case cannot occur because we assumed [see (3.1)] that there is at least one nonzero off-diagonal matrix element in each column (or row)

and the application of orthogonal transformations does not change this property. The process of clearing a matrix element on the first row and inflating the matrix may "accidentially" lead to $\alpha_1^{(n+1)} = 0$. Appendix B contains simple examples that illustrate the various cases. If $n + 1 < N$ there is no immediate danger for the method to break down. If there exists a permutation of the columns (and rows) $n + 1$ and n' $(n + 1 < n' \leq N)$ that yields $\alpha_1^{(n+1)} \neq 0$, we perform this permutation (in theory, not in practice of course) and continue as usual. However, if $n + 1 = N$ or if no such permutation exists, then the method has isolated an eigenvalue, but there is no guarantee that it is the smallest. In this case the matrix has been reduced to the form

$$H' = \begin{pmatrix} E' & 0 \\ 0 & X' \end{pmatrix}. \tag{3.25}$$

We have no other option than to repeat the procedure, i.e., isolate the smallest eigenvalue, for the remaining $(N-1)\times(N-1)$ matrix X'. However, according to the hypothesis made in the introduction, the number of relevant states N_R is assumed to be a small fraction of N. Hence $n \leq N_R \ll N$ and the case $\alpha_1^{(p)} = 0$, $n < p < N$ will hardly occur in practice.

3.4. Importance sampling algorithm

The approach outlined above is theoretically sound but useless for practical purposes. Indeed, in theory each inflation step is to be followed by an infinite number of plane rotations to transform the matrix to the form (3.15). Turning the method into a useful importance sampling algorithm only requires minor modifications. The theoretical discussion that follows does not address questions of efficiency. In the next chapter we show that the theoretical description of the method can be implemented as an efficient algorithm.

The order to annihilate the off-diagonal elements of the first row (and column) is fully determined by our desire to efficiently isolate an eigenvalue. Accordingly, the pair $(1, j)$ is chosen such that

$$|H_{1j}^{(n,m)}| = \max_{i>1} |H_{1i}^{(n,m)}| . \tag{3.26}$$

The first modification, identical to the one made in the case of the Jacobi method, is to limit the number of plane rotations for fixed n by introducing the threshold $\varepsilon_R^{(n,m)} > 0$. Rotations will be carried out if

$$|H_{1i}^{(n,m)}| \geq \varepsilon_R^{(n,m)}, \quad i = 2, \ldots, n . \tag{3.27}$$

or, in different words, until the size of all off-diagonal elements on the first row becomes smaller than the threshold $\varepsilon_R^{(n,m)}$. Keeping $\varepsilon_R^{(n,m)}$ fixed, the transformed matrix reads

$$H^{(n,m)} = \begin{pmatrix} E_1^{(n,m)} & \delta^{(n,m)\mathrm{T}} \\ \delta^{(n,m)} & \tilde{H} \end{pmatrix}, \qquad E_1^{(n)} \leq E_1^{(n,m)} , \quad \delta_i^{(n,m)} = H_{1i}^{(n,m)} , \quad i > 1 . \tag{3.28}$$

The difference between $E_1^{(n,m)}$ and $E_1^{(n)}$ can be estimated by invoking the monotonicity theorem (see refs. [1, 2] or appendix A). For the case at hand it implies

$$E_1^{(n,m)} - E_1^{(n)} \leq \| \delta^{(n,m)} \| = \left(\sum_{i>1} (H_{1i}^{(n,m)})^2 \right)^{1/2} < \sqrt{n} \varepsilon_R^{(n,m)} . \tag{3.29}$$

The second modification concerns the inflation step. It automatically provides a criterion to decide which states are important and which are not. Again we proceed by induction. Assume the number of important states is n. We pick a trial state $\hat{\phi}$ from the set of $N - n$ remaining states, for instance randomly. Recall [see eq. (3.1)] that there must be at least one nonzero element in the new row and column, a constraint which in practice will considerably reduce the set of states to choose from (see also chapter 5). We temporarily set $\phi_{n+1} = \hat{\phi}$, compute $\alpha_1^{(n+1,m)}$ and the corresponding change of the (1, 1) element [see eq. (3.5c)]

$$\Delta_{n+1}^{(n+1,m)} = \frac{2(H_{1n+1}^{(n,m)})^2}{H_{jj}^{(n+1,m)} - H_{11}^{(n+1,m)} + [(H_{n+1\,n+1}^{(n+1,m)} - H_{11}^{(n+1,m)})^2 + 4(H_{1n+1}^{(n+1,m)})^2]^{1/2}} . \tag{3.30}$$

If $\Delta_{n+1}^{(n+1,m)} \geq \varepsilon_A^{(n+1,m)}$ the trial state $\hat{\phi}$ is considered to be important and is added to the set of states. Clearly the threshold $\varepsilon_A^{(n+1,m)} > 0$ will control the importance sampling process. We set $\phi_{n+1} = \hat{\phi}$ and

$$U^{(n+1,k)} = \begin{pmatrix} U^{(n,k)} & 0 \\ 0 & 1 \end{pmatrix}, \quad k = 1, \ldots, m . \tag{3.31}$$

Unlike in the previous sections of this chapter, the plane rotation index is not reset to its initial value $m = 1$ when we inflate the matrices. Annihilation of the matrix element $\alpha_1^{(n+1,m)} = 0$ determines the new rotation matrix $U^{(n+1,m+1)}$. We finally replace n by $n + 1$, m by $m + 1$ and continue. If $\Delta_{n+1}^{(n+1,m)} < \varepsilon_A^{(n+1,m)}$, the trial state is rejected and a new trial state $\hat{\phi}$ is generated. If $\alpha_1^{(n+1,m)} = 0$, the trial state is always rejected.

In order to isolate the smallest eigenvalue the reduction has to be large enough. A sufficient condition can be derived by repeating the steps that led to (3.16). In place of (3.16) we have

$$H^{(n+1,m)} = \begin{pmatrix} E_1^{(n,m)} & \delta_2^{(n,m)} & \cdots & \delta_n^{(n,m)} & \alpha_1^{(n+1,m)} \\ \delta_1^{(n,m)} & \tilde{H}_{22} & \cdots & \tilde{H}_{2n} & \alpha_2^{(n+1,m)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \delta_n^{(n,m)} & \tilde{H}_{2n} & \cdots & \tilde{H}_{nn} & \alpha_n^{(n+1,m)} \\ \alpha_1^{(n+1,m)} & \alpha_2^{(n+1,m)} & \cdots & \alpha_n^{(n+1,m)} & H_{n+1\,n+1} \end{pmatrix}, \tag{3.32}$$

because now m is finite. Annihilating $\alpha_1^{(n+1,m)}$ leads to a matrix of the form (3.20). From (3.29) it is clear that if

$$\lambda_1 \leq E_1^{(n,m)} - \| \delta^{(n,m)} \| , \tag{3.33}$$

we will have $\lambda_1 < E_1^{(n)}$ (eq. 3.21). Repeating the reasoning that follows (3.23) establishes that if (3.33) is satisfied, the inflation step will guarantee covergence to the smallest eigenvalue. The condition for isolating the smallest eigenvalue follows from (3.33) and $E_1^{(n,m)} - \lambda_1 \geq \varepsilon_A^{(n,m)}$ and is given by

$$\varepsilon_A^{(n,m)} \geq \| \delta^{(n,m)} \| \,. \tag{3.34}$$

As usual with this kind of theoretical analysis, the bounds on the maximum reduction of the smallest eigenvalue may be too weak and, strict use of (3.34) may have a negative impact on the performance of the algorithm.

The construction of the importance sampling algorithm and the proof that it yields the smallest eigenvalue of H have now been completed. To summarize, there are three basic procedures. (1) For fixed n perform plane rotations involving pairs $(1, j)$ as long as $|H_{1,j}^{(n,m)}| > \varepsilon_R^{(n,m)}$, thereby increasing m. (2) Increase n by searching for a state that leads to a reduction of the (1, 1) element by more than $\varepsilon_A^{(n,m)}$. (3) Use a strategy, possibly tuned to the specific application, to decrease $\varepsilon_R^{(n,m)}$ and $\varepsilon_A^{(n,m)}$ in the course of the calculation.

3.5. Lower bounds to the lowest eigenvalue

The importance sampling algorithm described above yields the smallest eigenvalue of the matrix $H^{(N_R)}$. N_R is the final number of important states gathered during the inflation process. According to the separation theorem [1] this eigenvalue is an upper bound to the smallest eigenvalue of $H^{(N)}$. In practice N_R will be limited by the amount of CPU and memory a calculation is permitted to use. In general, we cannot expect to have any other knowledge about the effect of including the remaining $N - N_R$ states. Assuming that the importance sampling scheme works properly, the upper bound $E_1^{(N_R)}$ to the smallest eigenvalue E_1 of H is the best variational result one can ever hope to obtain using only N_R states. However, the result of an actual calculation will be $E_1^{(N_R,m)}$, an upper bound to $E_1^{(N_R)} = \lim_{m\to\infty} E_1^{(N_R,m)}$. We now wish to demonstrate that the importance sampling algorithm can be used to compute a lower bound to $E_1^{(N_R)}$ as well. We need

Theorem 3. Let λ_1 be the smallest eigenvalue of the real and symmetric matrix $A = \begin{pmatrix} x & y^T \\ y & Z \end{pmatrix}$. Then the smallest eigenvalue $\mu_1 = \mu_1(t)$ of the matrix $B(t) = Z - (x-t)^{-1} y y^T$, $x \neq t$ satisfies

$$\lambda_1 + \frac{\lambda_1 - t}{(x-t)(x-\lambda_1)} \| y^T v_1(\mu_1) \|^2 \leq \mu_1 \leq \lambda_1 + \frac{\lambda_1 - t}{(x-t)(x-\lambda_1)} \| y^T v_1(\lambda_1) \|^2 \,,$$

$$B(t) v_1(t) = \mu_1(t) v_1(t) \,. \tag{3.35}$$

The proof of theorem 3 is given in appendix A. Inequality (3.35) establishes that

$$\mu_1 \leq \lambda_1 \,, \quad \lambda_1 \leq t < x \quad \text{or} \quad \lambda_1 \leq \mu_1 \,, \quad t \leq \lambda_1 \,. \tag{3.36a, b}$$

In practice, result (3.36a) can be used as follows. Make a reasonable guess for t, for instance by setting $t = E_1^{(N_R,m)} < H_{11}$. Then construct the matrix

$$B^{(N_R)}(t) = \begin{pmatrix} H_{22} & H_{23} & \cdots & H_{2N_R} \\ H_{23} & H_{33} & \cdots & H_{3N_R} \\ \vdots & \vdots & \ddots & \vdots \\ H_{2N_R} & H_{3N_R} & \cdots & H_{N_R N_R} \end{pmatrix} - \frac{1}{H_{11} - t} \begin{pmatrix} H_{12}H_{12} & H_{13}H_{12} & \cdots & H_{1N_R}H_{12} \\ H_{12}H_{13} & H_{13}H_{13} & \cdots & H_{1N_R}H_{13} \\ \vdots & \vdots & \ddots & \vdots \\ H_{12}H_{1N_R} & H_{13}H_{1N_R} & \cdots & H_{1N_R}H_{1N_R} \end{pmatrix}, \tag{3.37}$$

and compute the smallest eigenvalue $\mu_1^{(N_R)}$ of this matrix. According to (3.36a)

$$\mu_1^{(N_R)} \leq E_1^{(N_R)} , \tag{3.38}$$

yielding the desired lower bound.

4. Implementation

The development of the SD method described in the previous chapter has largely been motivated by the idea that the method should exploit the "sparseness" of the solution rather than the fact that the matrix itself is sparse. Below we present our implementation of the SD algorithm and give a theoretical analysis of its operation count and memory usuage. In chapter 5 we show that this analysis is fully supported by actual calculations.

4.1. Stochastic diagonalization algorithm

In words, our implementation of the SD algorithm reads as follows:

(1) **Initialize data structure**
(2) **do**
(3) **if** {maximum of absolute value of off-diagonal elements of the first row is smaller than threshold for rejecting plane rotations}
(4) **then** generate a new trial state
(5) **if** {no important state has been found}
(6) **then** reduce the threshold(s)
(7) **else** inflate the matrix
end if
(8) **else** annihilate the off-diagonal element with the largest absolute value by performing a plane rotation
end if
(9) **end do**

The major parts of the algorithm are numbered 1 through 9 and are discussed in more detail below.

(1). An important step in the initialization procedure is to choose the first state ϕ_1. Obviously, the actual choice may heavily depend on the model (or matrix) at hand. The performance of the algorithm can be enhanced considerably by storing as many nonzero matrix elements of $H^{(n)}$ as possible. We will call this data structure the matrix element cache in what follows. It is expedient to arrange the matrix such that $H_{11} = \langle \phi_1 | H | \phi_1 \rangle$ is smaller than all other diagonal matrix elements. If that is not possible permutations of the kind described in section 3.3 may occur. For many physical models of interest there is a nontrivial relationship between the index i and the state ϕ_i itself (for an example, see chapter 5). Therefore some coding of the states is necessary. Typically the amount of storage needed for this list will increase linearly with N_R, the maximum number of important states the algorithm is allowed to use. Each time a trial state is generated we have to check that this trial state has not already been accepted as an important state. Therefore we have to scan through the list of important states to find out if the trial state is new or not. The CPU time for each search through this list can be reduced to a small fraction of the total CPU time by introducing a look-up table of length N_R that contains the indices of the important states, ordered according to some criterion. Evidently, all these data structures need to be initialized properly.

(2). The main loop goes as long as the available CPU time permits or, in more favourable cases, terminates if the number of important states reaches its maximum N_R.

(3). Here we decide whether an attempt is made to inflate the matrix or not. As long as the absolute value of at least one of the off-diagonal elements of the first row exceeds $\varepsilon_R^{(n,m)}$, we apply plane rotations without changing the size of the matrix. Recall that a plane rotation modifies two columns and two rows.

(4). In general, the procedure to generate trial states will heavily rely on specific features of the model (see chapter 5). Numerical experiments (see chapter 5) indicate that it is inefficient to generate one trial state at a time. It is more effective to generate N_t states and to select from this set the trial state that yields the largest reduction of the (1, 1) element. We denote the maximum reduction, encountered during N_t trials, by [see (3.30)]

$$\Delta^{(n+1,m)} = \max_{\hat{\phi} \in \{\hat{\phi}_1, \ldots, \hat{\phi}_{N_t}\}} \Delta_{n+1}^{(n+1,m)} . \tag{4.1}$$

If $\Delta^{(n+1,m)} \geq \varepsilon_A^{(n+1,m)}$ a new important state has been found.

A trial state may already belong to the set of important states $S^{(n)}$. A search through this list is necessary to find out if it does. If the search is successful, the trial state is rejected and a new one is generated. Assuming that the procedure generated a trial state that does not yet belong to the set $S^{(n)}$, the next step is to compute the new row of matrix elements [see (3.32)]. From (3.17c) it follows that this calculation is of the type

$$\sum_{i=1}^{n} (U^{(n,1)}(1, j_1, c_1, s_1) \cdots U^{(n,m)}(1, j_m, c_m, s_m))_{ji}^{\mathrm{T}} H_{i\,n+1}^{(n+1)} .$$

Here and in the remainder of this section we drop the first, for the present purposes, irrelevant, index of $j_{n,k}$, $c_{n,k}$ and $s_{n,k}$. In steps 1 and 7 we have already anticipated that we would need the first row of the product of plane rotation matrices $U^{(n,k)}(1, j_k, c_k, s_k)$, for $k \in \{1, \ldots, m\}$ so no extra work is involved here. What remains is to multiply this vector by the matrix $H^{(n)}$, the elements of

which hopefully reside in the matrix element cache. If some of the matrix elements are not in the cache, they have to be computed from scratch or fetched from disk. For simplicity we assume that getting a matrix element of H costs $\mathscr{M}$ arithmetic operations. In a worst-case situation the calculation of the new row of matrix elements takes $(2n-1)n\mathscr{M}$ arithmetic operations. Finally we also compute $H_{n+1\,n+1}$.

In the course of searching for important states, it may happen that the current value of the threshold $\varepsilon_{\mathrm{A}}^{(n,m)}$ is too large. Then none of N_{t} trial states qualifies as an important state. We allow such failures to occur N_{f} times. In practice $1 \leq N_{\mathrm{f}} \leq 100$. If after $N_{\mathrm{f}}N_{\mathrm{t}}$ attempts no new state has been added to the set $S^{(n)}$ then, at this stage, the generation of a new, important state has failed.

(5). If the search for a new, important state is succesful we inflate the matrix. Otherwise the thresholds will be reduced.

(6). The probability for finding a new important state can be increased by reducing the threshold $\varepsilon_{\mathrm{A}}^{(n,m)}$ and, if necessary, also $\varepsilon_{\mathrm{R}}^{(n,m)}$. In practice we simply divide $\varepsilon_{\mathrm{A}}^{(n,m)}$ by two and set $\varepsilon_{\mathrm{R}}^{(n,m)} = \varepsilon_{\mathrm{A}}^{(n,m)}$.

(7). Having found a new important state, most of the data structures have to be updated. Obviously the new state itself has to be added to the list $S^{(n)}$. For computational efficiency, the current values of the diagonal matrix elements $H_{ii}^{(n,m)}$ for $i \in \{1, \ldots, n\}$, the plane rotation coefficients (c_{j_k}, s_{j_k}) and the indices j_k, for $k \in \{1, \ldots, m\}$ are always in memory. Clearly, since $n \leftarrow n+1$, all these lists have to be extended accordingly. If the mapping of an index i to a state ϕ_i requires an additional array (see part 1) the look-up tables that are used to search the list of states also need to be updated. If use is made of a matrix element cache and if memory is not exhausted, the matrix elements $\langle \phi_n | H | \phi_i \rangle$, for $i \in \{1, \ldots, n\}$ that are not zero should be copied to the cache. All these operators take a neglegible fraction of CPU time.

More elaborate is the calculation of the new values of the off-diagonal elements of the first row $H_{1i}^{(n,m)}$, for $i \in \{2, \ldots, n\}$. First we calculate $\mathscr{U}_{1i}^{(n,m)}$, for $i \in \{1, \ldots, n\}$ by multiplying the plane rotations (c_{j_k}, s_{j_k}) where $k \in \{1, \ldots, m\}$. This takes $6m$ arithmetic operations. We keep this vector in storage to speed up the calculation of a new row (column) of the matrix (see part 4). Then we multiply the matrix $H^{(n)}$ by this vector (i.e., we form $x^{\mathrm{T}}A$, not Ax). The second step takes $\mathscr{M}n(2n-1)$ operations if the matrix is completely full. Otherwise it takes less. Finally we multiply the resulting row vector by the matrix $\mathscr{U}_{ij}^{(n,m)}$, for $i, j \in \{1, \ldots, n\}$, not by first computing this matrix and performing the multiplication afterwards (that would take n^2 operations), but by multiplying the row vector by the sequence of plane rotations. Also this step costs $6m$ arithmetic operations. For step 3 we need the index j of the off-diagonal element of the first row with the largest absolute value. This search takes $n\mathscr{S}$ operations where $\mathscr{S}$ is of order one. In total, the number of arithmetic operations performed in step 7 is, to a good approximation, given by $n(2n-1)\mathscr{M} + 12m + n\mathscr{S}$. Numerical experiments demonstrate that $m = \mathcal{O}(n)$ so that for large n, the computation time of step 7 will increase quadratically with n.

(8). Annihilation of an off-diagonal element of the first row consists of two basic steps. Step 7 provides the necessary information to find the index j of the element in question. The matrix elements $H_{11}^{(n,m)}$, $H_{jj}^{(n,m)}$ and $H_{1j}^{(n,m)}$ are always in memory and determine the plane rotation $(c_{j_{m+1}}, s_{j_{m+1}})$ that annihilates the element $(1, j_{m+1})$. This calculation takes very little CPU time. Then we add this plane rotation to the list $(m \leftarrow m+1)$ and compute all the off-diagonal elements of the first row. This calculation is identical to the one performed in step 7. For step 3 we again need the index j of the off-diagonal element of the first row with the largest absolute value. As in step 7, for large n the computation time of step 8 will increase quadratically with n.

(9). Having determined n important states, we can reconstruct the approximation to the ground state from the sequence of plane rotations and compute all physical properties of interest.

From the above description of the SD algorithm it follows that memory usage increases linearly with N_R, the maximum number of important states the SD algorithm will search for. Assuming the importance sampling is working properly, the CPU time increases quadratically with the number of important states n.

4.2. Demonstration programs

In this subsection we present elementary FORTRAN programs that serve to illustrate the theoretical ideas of sections 3.2, 3.3 and 3.5. They can be used to carry out small-scale numerical experiments and are not intended to be used for application such as the one presented in section 5. A listing of the main program is given in display 1 below. As an example we take a tri-diagonal matrix having eigenvalues

$$E_i = 4 \sin^2[i\pi/(N+1)] , \quad i \in \{1, \ldots, N\} . \tag{4.2}$$

The code that generates this matrix is given in display 2. The piece of code in display 3 constructs the matrix B of section 3.5 [see (eq. 3.37)]. In display 4 comes a routine that implements the cyclic Jacobi method. An algorithm for the modified Jacobi scheme is given in display 5. Note that the code to determine the plane rotation differs from the code in JACOBI1. In both cases care has been taken that the calculation of the rotation sine and cosine is accurate under all circumstances. To keep the JACOBI2 procedure as simple as possible we do not allow for permutations of the kind described in section 3.3.

Typical output of the main program looks like display 6. According to (3.36b) the above lower bound could be used as the new value for t, yielding an upper bound to the lowest eigenvalue.

5. Application

5.1. Model

The method outlined above has been applied to the two-dimensional single-band Hubbard model, a reference system for testing methods for simulating fermions [16]. It is assumed to describe some of the essential features of strongly correlated electron systems, and has attracted considerable attention in connection with the high-temperature superconductors [17]. The model is defined by the Hamiltonian

$$H = -t \sum_{\langle i,j \rangle, \sigma} (c^+_{i,\sigma} c_{j,\sigma} + c^+_{j,\sigma} c_{i,\sigma}) + U \sum_i n_{i\downarrow} n_{i\uparrow} , \tag{5.1}$$

where $c^+_{i,\sigma}$ ($c_{i,\sigma}$) creates (annihilates) a fermion of spin $\sigma = \uparrow, \downarrow$ at site i, t is the hopping matrix element, U represents the on-site Coulomb interaction strength, and the sum over i and j is

```
      implicit real*8 (a-h,o-z)
      parameter (ndim=16)
      real*8 H(ndim,ndim),B(ndim,ndim)

1     write(6,*)' '
      read(5,*,err=1,end=10000) n,eps,nsweep
      write(6,*)' ** Cyclic Jacobi method **'
      write(6,*)' '
      write(6,*) '  Matrix dimension:',n
      write(6,*) '  Threshold:',eps
      write(6,*) '  Maximum number of sweeps:',nsweep
      call matrix(ndim,n,H)
      call jacobi1(H,ndim,n,nsweep,eps,nzero,niter)
      write(6,*)'  Number of non-diagonal entries > EPS: ',nzero
      write(6,*)'  Total number of Jacobi rotations: ',niter
      write(6,'(/'' Index, All eigenvalues:''/1x,23(''-''))')
      write(6,'(1x,i5,e15.5)') (i,H(i,i),i=1,n)

2     write(6,*)' '
      read(5,*,err=2,end=10000) n,eps,nsweep
      write(6,*)' ** Modified Jacobi method **'
      write(6,*)' '
      write(6,*) '  Matrix dimension:',n
      write(6,*) '  Threshold:',eps
      write(6,*) '  Maximum number of sweeps:',nsweep
      call matrix(ndim,n,H)
      ne=1
      call jacobi2(H,ndim,n,ne,nsweep,eps,nzero,niter)
      write(6,*)'  Number of non-diagonal entries > EPS: ',nzero
      write(6,*)'  Total number of Jacobi rotations: ',niter
      write(6,'(/'' Index, Eigenvalues:''/1x,19(''-''))')
      write(6,'(1x,i5,e15.5)') (i,H(i,i),i=1,ne)

3     write(6,*)' '
      read(5,*,err=3,end=10000) n,eps,nsweep,t
      write(6,*)' ** Lower bound using modified Jacobi method **'
      write(6,*)' '
      write(6,*) '  Matrix dimension:',n
      write(6,*) '  Threshold:',eps
      write(6,*) '  Maximum number of sweeps:',nsweep
      write(6,*) '  t according to Theorem 3:',t
      call matrix(ndim,n,H)
      call matrix2(ndim,n,H,B,t)
      ne=1
      call jacobi2(B,ndim,n-1,ne,nsweep,eps,nzero,niter)
      write(6,*)'  Number of non-diagonal entries > EPS: ',nzero
      write(6,*)'  Total number of Jacobi rotations: ',niter
      write(6,'(/'' Index, Lower bound on smallest eigenvalue:''/
     1       1x,42(''-''))')
      write(6,'(1x,i5,e15.5)') (i,B(i,i),i=1,ne)

10000 end
```

Display 1. FORTRAN listing of the main program.

restricted to nearest neighbors. The linear size in the x (y) direction will be denoted by L_x (L_y). As usual we adopt periodic boundary conditions.

In order to minimize the computational effort, it is expedient to choose the most appropriate representation of the states of the system. The optimal choice depends on model parameters such as

```
      subroutine matrix(ndim,n,H)
c
c Example: tridiagonal matrix
c
      implicit real*8 (a-h,o-z)
      real*8 H(ndim,ndim)
      do j=1,n
       do i=1,n
        h(i,j)=0
       enddo
      enddo
      do i=1,n-1
       h(i,i+1)=-1
       h(i+1,i)=-1
      enddo
      do i=1,n
       h(i,i)=2
      enddo
      return
      end
```

Display 2. FORTRAN listing of the subroutine generating the matrix with the eigenvalues of eq. (4.2).

```
      subroutine matrix2(ndim,n,H,B,t)
c
c Construct matrix (3.38)
c
      implicit real*8 (a-h,o-z)
      real*8 H(ndim,ndim),B(ndim,ndim)
      if(H(1,1).ne.t) then
       do j=1,n-1
        do i=1,n-1
         B(i,j)=H(i+1,j+1)-H(1,i+1)*H(1,j+1)/(H(1,1)-t)
        enddo
       enddo
      else
       stop' MATRIX2: t=H(1,1) is not allowed !'
      endif
      return
      end
```

Display 3. FORTRAN listing of the subroutine constructing matrix B of eq. (3.37).

U/t and the electron filling. We have chosen to work in the wave-number representation instead of the real-space one. This automatically accounts for the translational symmetry. The states ϕ_i are Slater determinants built from plane waves. In this representation the matrix elements $\langle\phi_i|H|\phi_j\rangle$ are simple and, using proper bit-coding techniques, can be computed efficiently.

The number of states can be reduced further by exploiting the full symmetry of the square lattice. Hamiltonian (5.1) is invariant under operation O_R of elements R of the point group $G = C_{4v}$ of the square lattice, i.e.,

$$[O_R, H] = 0\,. \tag{5.2}$$

```
      subroutine jacobi1(H,ndim,n,nsweep,eps,nzero,niter)
c diagonalization of N * N matrix H(.,.), using serial Jacobi
c Input:  H(.,.) matrix. Only the upper triangular part is used and modified
c         NDIM : leading dimension of H(.,.)
c         N : actual dimension
c         NSWEEP : maximum number of lattice sweeps
c         EPS : Threshold for Jacobi rotations
c OUTPUT: H(.,.) diagonal matrix
c         NZERO : Number of non-diagonal entries > EPS (should be zero)
c         NITER : Total number of Jacobi rotations
      implicit real*8(a-h,o-z)
      real*8 H(ndim,ndim)
      niter=0
      do l=1,nsweep
       nzero=0
       do i=1,n-1
        do j=i+1,n
         x12=H(i,j)
         if(abs(x12).gt.eps) then ! one plane rotation around (i,j)
          nzero=nzero+1
          x11=H(i,i)
          x22=H(j,j)
          r0=(x22-x11)/(2*x12)
          r1=abs(r0)
          if(r1.le.1.e30) then
           t=1/(r1+sqrt(r1*r1+1))
          else
           t=1/(2*r1)
          endif
          if(r0.lt.0) t=-t
          c=1/sqrt(t*t+1)
          s=t*c
          H(i,i)=x11-t*x12
          H(j,j)=x22+t*x12
          H(i,j)=0
          do k=1,i-1
           r0=H(k,i)
           H(k,i)=c*r0-s*H(k,j)
           H(k,j)=c*H(k,j)+s*r0
          enddo
          do k=i+1,j-1
           r0=H(i,k)
           H(i,k)=c*r0-s*H(k,j)
           H(k,j)=c*H(k,j)+s*r0
          enddo
          do k=j+1,n
           r0=H(i,k)
           H(i,k)=c*r0-s*H(j,k)
           H(j,k)=c*H(j,k)+s*r0
          enddo
         endif
        enddo
       enddo
       if(nzero.le.0) return
       niter=niter+nzero
      enddo
      return
      end
```

Display 4. FORTRAN listing of the subroutine implementing the cyclic Jacobi method.

```
      subroutine jacobi2(H,ndim,n,ne,nsweep,eps,nzero,niter)
c (partial) diagonalization of N * N matrix H(.,.), using modified Jacobi
c Input:  H(.,.) matrix. Only the upper triangular part is used and modified
c         NDIM : leading dimension of H(.,.)
c         N : actual dimension
c         nE : number of desired eigenvalues. 0 < nE < N+1
c         NSWEEP : maximum number of lattice sweeps
c         EPS : Threshold for Jacobi rotations
c OUTPUT: H(.,.) diagonal matrix
c         NZERO : Number of non-diagonal entries > EPS (should be zero)
c         NITER : Total number of Jacobi rotations
      implicit real*8(a-h,o-z)
      real*8 H(ndim,ndim)
      niter=0
      do l=1,nsweep
       nzero=0
       do i=1,max(min(ne,n-1),1)
        y12=0
        do k=i+1,n ! find the largest element
         h12=H(i,k)
          if(abs(h12).gt.y12) then
           j=k
           y12=abs(h12)
          endif
        enddo
        if(y12.gt.eps) then ! diagonalize the 2x2 matrix (see Lemma 1)
         nzero=nzero+1
         h11=H(i,i)
         h22=H(j,j)
         h12=H(i,j)
         r1=H22-H11
          if(r1.gt.0) then
           r1=r1/2
           r2=sqrt(r1*r1+H12*H12)
           r1=-h12/(r2+r1)! t                                    (a)
           r4=(H22+H11)/2
           h(i,i)=r4-r2
           h(j,j)=r4+r2
           r3=1/sqrt(1+r1*r1)
           c=r3
           s=r1*r3
```

Display 5. FORTRAN listing of the subroutine implementing the modified Jacobi method.

The symmetry operation O_R rotates the wave vectors, characterizing the Slater determinants of plane-wave single-particle states. In general, translations and rotations do not commute [18]. The relevant point group G_Q is the subgroup of G that transforms the total wave vector Q into Q modulo a reciprocal lattice vector [18]. The eigenstates of the Hamiltonian H can be classified according to the irreducible representations $D^{(\mu)}$ of G_Q [19]. Let $\chi^{(\mu)}(R)$ denote the character of group element $R \in G_Q$ in the representation $D^{(\mu)}$. Symmetry-adapted functions $\phi_i^{(\mu)}$ are constructed according to [19]

$$\phi_i^{(\mu)} = (1/\mathcal{N}_{\phi_i}) \sum_{R \in G_Q} \chi^{(\mu)}(R) O_R \phi_i \,, \tag{5.3}$$

```
      else if(r1.lt.0) then
       r1=r1/2
       r2=sqrt(r1*r1+h12*h12)
       r1=h12/(r2-r1)! t
       r4=(H22+H11)/2
       h(i,i)=r4+r2
       h(j,j)=r4-r2
       r3=1/sqrt(1+r1*r1)  ! 1/sqrt(1+t*t)
       c=r3
       s=r1*r3
      else
       if(h12.gt.0) then
        h(i,i)=H11-h12
        h(j,j)=H22+h12
        r1=2
        r1=1/sqrt(r1)
        c=r1
        s=-r1
       else
        h(i,i)=H11+h12
        h(j,j)=H22-h12
        r1=2
        r1=1/sqrt(r1)
        c=r1
        s=r1
       endif
      endif
     H(i,j)=0
     do k=1,i-1 ! update the two rows and columns
      r0=H(k,i)
      H(k,i)=c*r0-s*H(k,j)
      H(k,j)=c*H(k,j)+s*r0
     enddo
     do k=i+1,j-1
      r0=H(i,k)
      H(i,k)=c*r0-s*H(k,j)
      H(k,j)=c*H(k,j)+s*r0
     enddo
     do k=j+1,n
      r0=H(i,k)
      H(i,k)=c*r0-s*H(j,k)
      H(j,k)=c*H(j,k)+s*r0
     enddo
    endif                                              (b)
   enddo
   if(nzero.le.0) return
   niter=niter+nzero
  enddo
  return
  end
```

Display 5. (continued).

where ϕ_i is some state and $\mathcal{N}_{\phi_i}$ is a constant determined by the normalization condition $\langle \phi_i^{(\mu)} | \phi_i^{(\mu)} \rangle = 1$. Since $\langle \phi_i^{(\mu)} | H | \phi_j^{(\nu)} \rangle = 0$ if $\mu \neq \nu$, the matrix H decomposes into blocks corresponding to the different symmetries [19]. The matrix elements of the Hamiltonian with respect to the symmetry adapted functions (5.3) are obtained as

```
 ** Cyclic Jacobi method **

  Matrix dimension:           4
  Threshold:  9.9999999999999999E-04
  Maximum number of sweeps:         100
  Number of non-diagonal entries > EPS:            0
  Total number of Jacobi rotations:           14

Index, All eigenvalues:
-----------------------
    1    0.36180E+01
    2    0.38197E+00
    3    0.13820E+01
    4    0.26180E+01

 ** Modified Jacobi method **

  Matrix dimension:           4
  Threshold:  9.9999999999999999E-04
  Maximum number of sweeps:         100
  Number of non-diagonal entries > EPS:            0
  Total number of Jacobi rotations:           17

Index, Eigenvalues:
-------------------
    1    0.38197E+00

 ** Lower bound using modified Jacobi method **

  Matrix dimension:           4
  Threshold:  9.9999999999999999E-04
  Maximum number of sweeps:         100
  t according to Theorem 3:  0.4000000000000000
  Number of non-diagonal entries > EPS:            0
  Total number of Jacobi rotations:            8

Index, Lower bound on smallest eigenvalue:
------------------------------------------
    1    0.37903E+00
```

Display 6. Example of the output.

$$\langle\phi_i^{(\mu)}|H|\phi_j^{(\mu)}\rangle = \frac{1}{\mathcal{N}_{\phi_i}\mathcal{N}_{\phi_j}} \sum_{S\in G_Q} \sum_{R\in G_Q} \chi^{(\mu)}(R)^* \chi^{(\mu)}(S) \langle\phi_i|O_R^\dagger H O_S|\phi_j\rangle \,, \tag{5.4a}$$

$$= \frac{1}{\mathcal{N}_{\phi_i}\mathcal{N}_{\phi_j}} \sum_{T\in G_Q} \sum_{R\in G_Q} \frac{\chi^{(\mu)}(R)^*\chi^{(\mu)}(RT)}{\chi^{(\mu)}(T)} \chi^{(\mu)}(T) \langle\phi_i|HO_T|\phi_j\rangle \,, \tag{5.4b}$$

$$= \frac{1}{\mathcal{N}_{\phi_i}\mathcal{N}_{\phi_j}} \sum_{T\in G_Q} g_Q^{(\mu)}(T)\chi^{(\mu)}(T) \langle\phi_i|HO_T|\phi_j\rangle \,, \tag{5.4c}$$

$$g_Q^{(\mu)}(T) = \sum_{R\in G_Q} \frac{\chi^{(\mu)}(R)^*\chi^{(\mu)}(RT)}{\chi^{(\mu)}(T)} \,. \tag{5.4d}$$

It is more efficient to use (5.4c, d) instead of (5.4a). Unless explicitly specified otherwise, the results presented have been computed for $Q = (0, 0)$ and the A_1 representation of $G_{Q=0} = G = C_{4v}$.

5.2. *Performance of the SD algorithm*

The purpose of this section is to discuss the performance of the SD algorithm, using 2D Hubbard model (5.1) as an example. The results of the numerical experiments presented below give support to the theoretical performance analysis given earlier. Evidently, significant parts of the code that implements the SD algorithm for this particular Hamiltonian will be model-specific. They can be optimized by exploiting all features of the model as much as possible. Although this optimization helps to reduce the actual CPU time required to carry out a calculation, it has minor impact on the performance analysis. It merely changes the actual numbers, not their dependence on variables such as the number of states, plane rotations, etc.

A more delicate question is whether the matrix H, representing model (5.1), is a "typical" case. We cannot give a complete answer to this, but stress that it certainly is a "difficult" case in the sense that the model parameters (U/t, electron filling) will be chosen such that QMC calculations of ground-state properties encounter severe difficulties or even fail [16].

A crucial step in the SD algorithm is the generation of the trial states $\hat{\phi}$. Clearly, this is the most model-specific part of the algorithm. The efficiency of the SD algorithm, as well as of other algorithms, can be improved by incorporating as much knowledge about the system as possible. As mentioned above, we have chosen to adopt the wave-number representation. The first state is taken to be the Fermi sea

$$|\text{Fermi sea}\rangle = c^+_{k_1,\uparrow} \cdots c^+_{k_{N_\uparrow},\uparrow} c^+_{q_1,\downarrow} \cdots c^+_{q_{N_\downarrow},\downarrow} |0\rangle \,, \tag{5.5}$$

where $N_\uparrow$ and $N_\downarrow$ are the numbers of electrons with spin up and down, respectively. Selecting the total wave number Q leads to the constraint $\Sigma_i^{N_\uparrow} k_i + \Sigma_j^{N_\downarrow} q_j = Q$. Minimizing $\langle \text{Fermi sea} | H | \text{Fermi sea} \rangle$, i.e., solving the Hartree–Fock problem for the Hubbard model for fixed Q, determines the wave numbers k_i and q_j entering (5.5). For electron fillings corresponding to an open-shell situation there will be more than one Fermi sea. We simply take one of them. As an option, the procedure outlined at the end of the previous section can be used to exploit the rotational symmetry. This completes the construction of the state ϕ_1.

Trial states $\hat{\phi}$ are generated as follows. A random process selects one of the states from the set $S^{(n)}$, say ϕ_i, $i \in \{1, \ldots, n\}$. In the wave-number representation the Hubbard model reads

$$H = \sum_{k,\sigma} \varepsilon_k c^+_{k,\sigma} c_{k,\sigma} + \frac{U}{L} \sum_{k,p,q} c^+_{k+q,\uparrow} c_{k,\uparrow} c^+_{p-q,\downarrow} c_{p,\downarrow} \,, \tag{5.6}$$

where L is the total number of lattice sites. Accordingly, for $\hat{\phi} \neq \phi_i$, $\langle \hat{\phi} | H | \phi_i \rangle \neq 0$ if and only if $\hat{\phi}$ and ϕ_i differ by two particle–hole excitations. This implies that a trial state must be of the form

$$|\hat{\phi}\rangle = c^+_{k+q,\sigma} c_{k,\sigma} c^+_{p-q,\sigma'} c_{p,\sigma'} |\phi_i\rangle \,, \tag{5.7}$$

whereby $\sigma, \sigma' = \uparrow, \downarrow$ and the wave numbers (k, p, q) have to be chosen such that $\hat{\phi} \neq 0$. Usually there are many sets $(k, p, q, \sigma, \sigma')$ for which $\hat{\phi} \neq 0$ and we use a Monte Carlo process to pick out

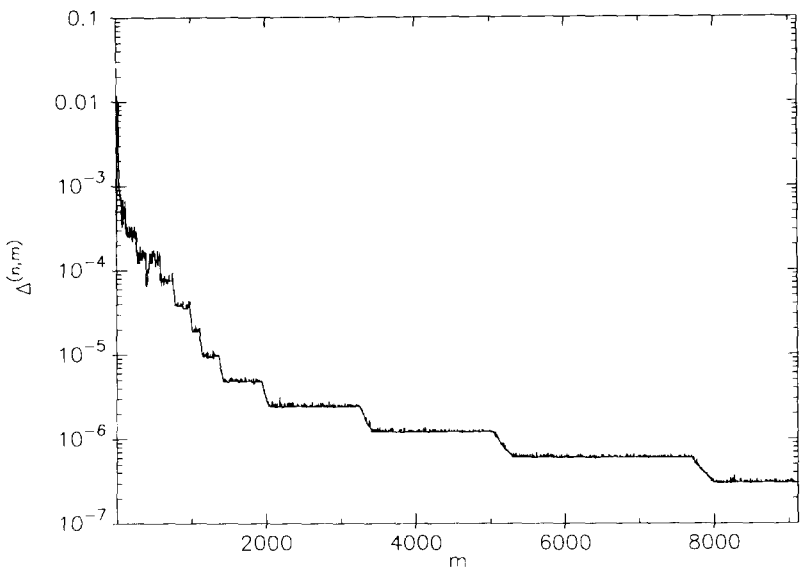

Fig. 1. Reduction $\Delta^{(n,m)}$ of $E_1^{(n,m)}$ as a function of the number of plane rotations m. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 5$ and $U/|t| = 4$.

a particular one. Care must be taken that this process can generate all allowed sets. This recipe to generate trial states ensures that condition (3.1) is satisfied. Again, rotational symmetry can be used to project out from this trial state, a trial state that has the proper transformation properties.

A first impression of the performance of the SD algorithm can be obtained by looking at the decrease of the (1, 1) matrix element. Figure 1 shows the maximum reduction $\Delta^{(n,m)}$ of $E_1^{(n,m)}$ as a function of the number of rotations m. The stepwise decrease of the threshold(s) is clearly reflected in the data. The SD algorithm is working properly if the number of rotations m does not increase much faster than the number of important states n. Figure 2 reveals that piecewise, m increases linearly with n, indicating that the SD algorithm is performing well. After each change of $\varepsilon_A^{(n,m)}$, the SD algorithm carries out some rotations before it attempts to add a new state, leading to the small steps in the data. A typical plot of the CPU time (as measured on a CRAY Y-MP4/64 in single processor mode) as a function of n is given in fig. 3. In general we find that the CPU time increases with n^2 for large n. The most time-consuming part of the calculation is to compute, for each of the N_t trial states (see chapter 4), the corresponding column $(\alpha_1^{(n+1,m)}, \ldots, \alpha_n^{(n+1,m)})$ [see eqs. (3.16) and (3.17)] of the matrix.

The size of the pool of trial states N_t can be used to tune the algorithm. Let us fix the maximum number of important states ($N_R = 1000$ for the results shown in fig. 4–6) and examine the N_t dependence of $E_1^{(n,m)}$ (for m large, effectively $m \to \infty$) and of the CPU time required to obtain $E_1^{(n,m)}$. From fig. 4 it is clear that for N_t too small ($N_t \ll 50$) the 1000 states found by the SD method are not very optimal. On the other hand, taking N_t too large does not bring substantial improvements. From the earlier discussion the CPU time is expected to scale linearly with N_t. From fig. 4 it follows that there must be an optimal choice of N_t. Acceptance of unimportant states can be prevented by taking N_t large, whereas the CPU time can be reduced by chosing N_t small. A plot of the CPU time required to obtain a fixed value of $E_1^{(n,m)}$ as a function of N_t gives a good indication of the optimal value of N_t. This is illustrated in fig. 5. For the case at hand $N_t = 30$ would be a good choice.

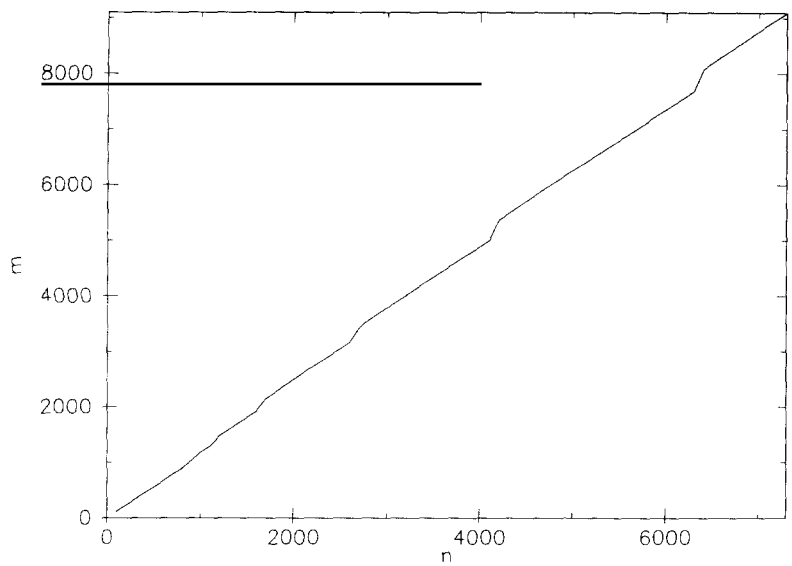

Fig. 2. Number of rotations m as a function of number of states n. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 5$ and $U/|t| = 4$

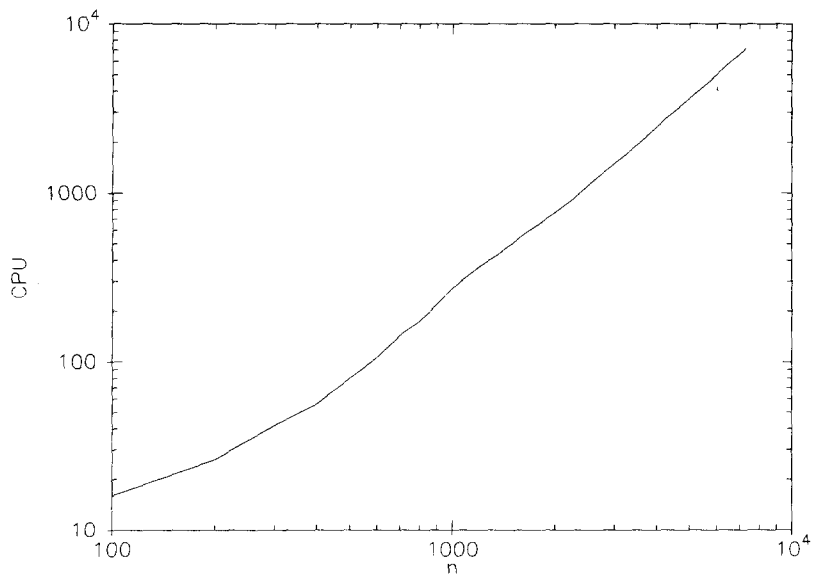

Fig. 3. CPU time (in seconds) to carry out the calculations of figs. 1, 2.

Valuable information about the importance sampling process is contained in the distribution of weights of the states ϕ_j to the (approximate) ground state $\Phi_1^{(n,m)} = \Sigma_{j=1}^{n} \mathcal{U}_{j1}^{(n,m)} \phi_j$ [see also (3.13)]. This distribution is given by

$$d^{(n,m)}(x) = \frac{1}{2\Delta x} \sum_{i=1}^{n} \Theta(\Delta x + [\mathcal{U}_{i1}^{(n,m)}]^2 - x)\Theta(x + \Delta x - [\mathcal{U}_{i1}^{(n,m)}]^2) , \tag{5.8}$$

$$\Theta(x) = 1 , \quad x \geq 0 ; \qquad \Theta(x) = 0 , \quad x < 0 , \tag{5.9}$$

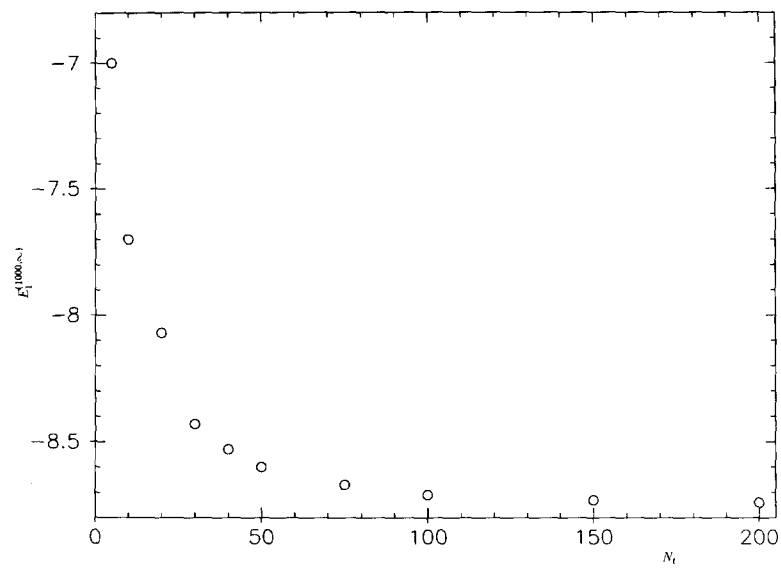

Fig. 4. $E_1^{(1000,\infty)}$ as a function of the size of the pool of trial states N_t. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 7$ and $U/|t| = 4$.

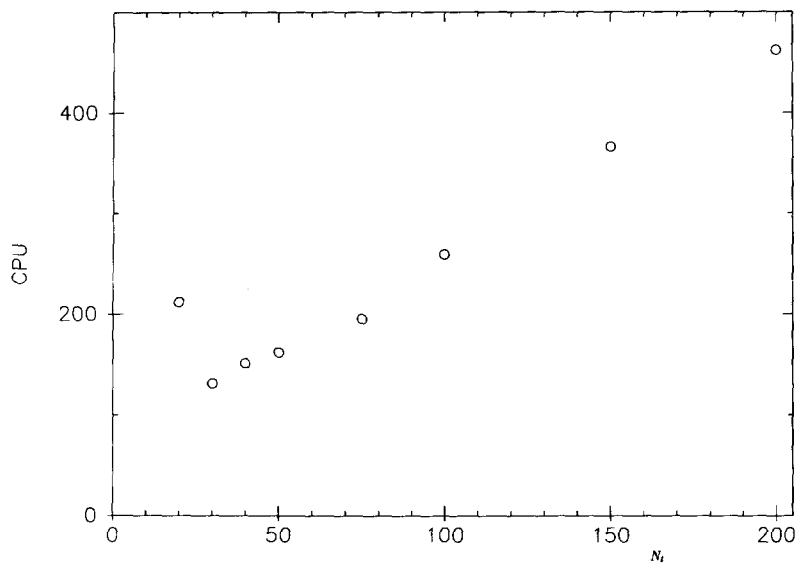

Fig. 5. CPU time (in seconds) used to reach the approximate ground-state energy $E_1^{(n,m)} = -15.1$ as a function of the number of Monte Carlo trials N_t. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 7$ and $U/|t| = 4$.

where $2\Delta x$ is the bin size of the histogram. By construction (5.8) is normalized to number of states, i.e.,

$$\int_0^1 d^{(n,m)}(x)\mathrm{d}x = n \,. \tag{5.10}$$

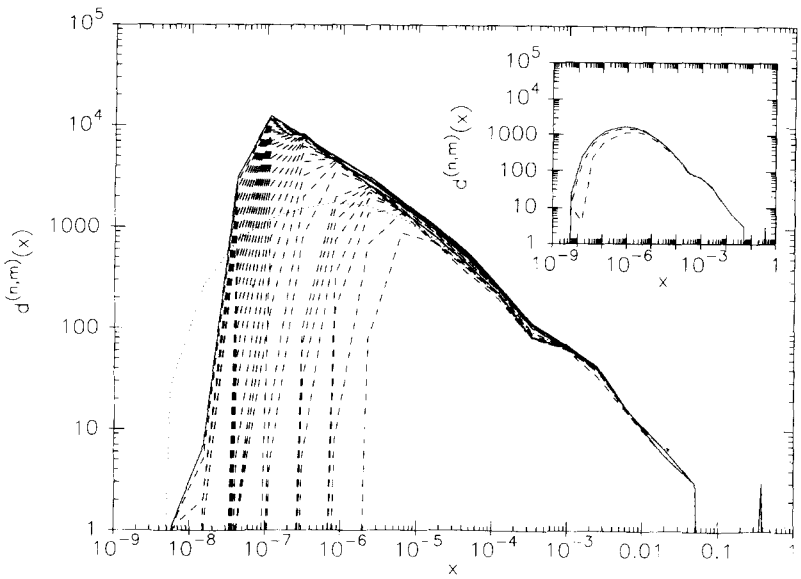

Fig. 6. Distribution $d^{(n,m)}(x)$ of the weights of the states ϕ_i as obtained from the approximate ground-state wave function $\Phi_1^{(n,m)}$ for increasing sizes of the set of states $S^{(n)}$. The solid line gives the result for $n = 3 \times 10^4$. The dashed lines represent distributions for various $n < 3 \times 10^4$, approaching the solid line as n increases. The size of the pool of trial states $N_t = 40$. The dotted line is the distribution for $n = 10^4$ and $N_t = 5$. The inset shows the results for $n \leq 10^4$ and $N_t = 5$. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 7$ and $U/|t| = 4$.

A typical example of a distribution $d^{(n,m)}(x)$ for various n is depicted in fig. 6. Comparing the final distribution (solid line) with intermediate ones (dashes lines) it is clear that new states mainly appear in the low-weight part of the distribution. This is a strong indication that the algorithm is highly efficient in collecting the most important contributions to the ground state. Moreover, fig. 6 demonstrates that states considered to be important at an early stage of the calculation, remain important at later stages. Accordingly, for the matrices at hand, there seems to be no need for an additional device to disregard accepted states.

Above we already argued that the choice of N_t affects the efficiency of the SD scheme. This is once more illustrated in the inset of fig. 6 where we show the results of a calculation with $N_t = 5$ instead of $N_t = 40$ (also compare the solid and dotted lines of fig. 6). Although the limiting distributions for $x \geq 10^{-5}$ are similar, the $N_t = 5$ run collects additional states with weights distributed over a much larger x interval. This is clear evidence that the SD algorithm tends to accumulate less important states if N_t is too small. The pool of trial states from which a new state is chosen must be large enough to ensure good performance.

In chapter 3 we gave a proof that the modified Jacobi process reduces to zero all off-diagonal elements on the first row and column. Figure 7 shows a representative plot of the norm of the off-diagonal elements $\| \delta^{(n,m)} \|$ [see (3.28)] as a function of the number of rotations m. The maximum number of important states has been fixed to $N_R = 400$. Initially, for small m, the size of the off-diagonal elements changes rapidly as a function of m. This is because when large (in absolute value with respect to $\varepsilon^{(n,m)}$) off-diagonal elements are annihilated, other large elements appear. Once the target number of relevant states N_R has been reached $\| \delta^{(N_R,m)} \|$ vanishes exponentially with m (see inset of fig. 7). The classical Jacobi method also displays this behavior [1].

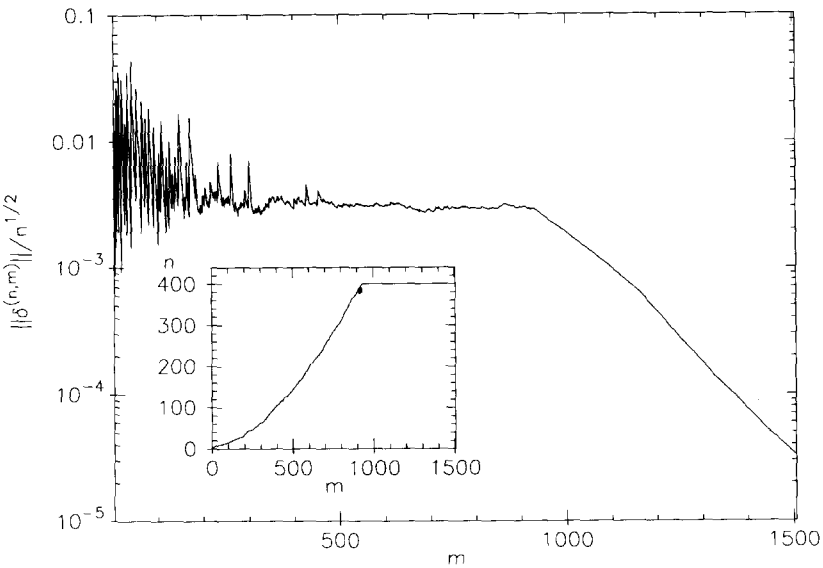

Fig. 7. The length of the vector of off-diagonal elements on the first row $\| \delta^{(n,m)} \|/n^{1/2}$ as a function of the number of plane rotations m. The inset shows the number of states n as a function of the number of plane rotations m. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 5$ and $U/|t| = 4$.

In general the symmetry of the ground state is not known in advance. The use of symmetry-adapted states effectively block-diagonalizes the matrix. The ground state and its symmetry properties follow from the calculation of the smallest eigenvalues and corresponding eigenvectors of each of the submatrices. The CPU time required to generate symmetry-adapted trial states and to compute the matrix element $\langle \hat{\phi} | H | \phi_i \rangle$ is substantially larger than when no use is made of the full rotational symmetry. However, in the latter case, the SD has to collect more states and, compared to the former, has to perform more rotations to achieve the same level of accuracy. The extra work to account for the rotational symmetry is more than just compensated for by the fact that in the end less states and rotations are needed. This expectation is supported by the data presented in figs. 8, 9, where we show the relative deviation of the approximate ground-state energy as a function of the number of states n. Our calculations suggest that to reach a certain accuracy, the number of unsymmetrized states is roughly a factor 16 (i.e., the number of symmetry operations of the square + spin interchange) larger than the number of symmetry-adapted states.

5.3. *Results*

We now discuss results for two-dimensional square Hubbard systems of various sizes, fillings and interactions U ($t = 1$ in our numerical work) and compare SD results for various physical quantities with exact numerical diagonalization and PQMC data. The SD results presented below support the general consensus about the nature of the ground state of the 2D Hubbard model [16] and add little to the present understanding of the properties of the Hubbard model.

A detailed description of PQMC methods can be found elsewhere [16]. The PQMC method we have used in this work is a variant of the technique used by Sorella et al. to study the Hubbard model [20]. Instead of continuous variables we have used Ising spins to represent the auxillary variables entering the Hubbard–Stratonovich transformation. We have employed the Metropolis

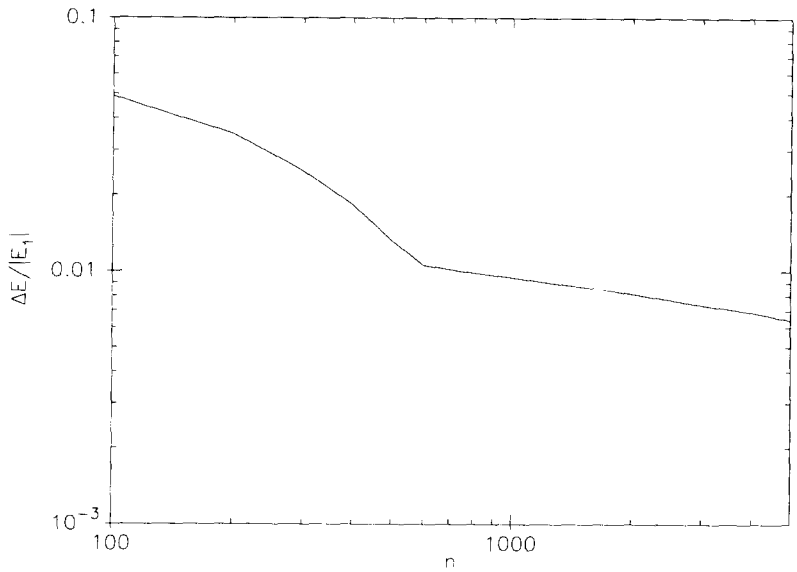

Fig. 8. Relative deviation $\Delta E/|E_1| = (E_1^{(n,m)} - E_1)/|E_1|$ as a function of the number of unsymmetrized states n. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 4$ and $U/|t| = 4$.

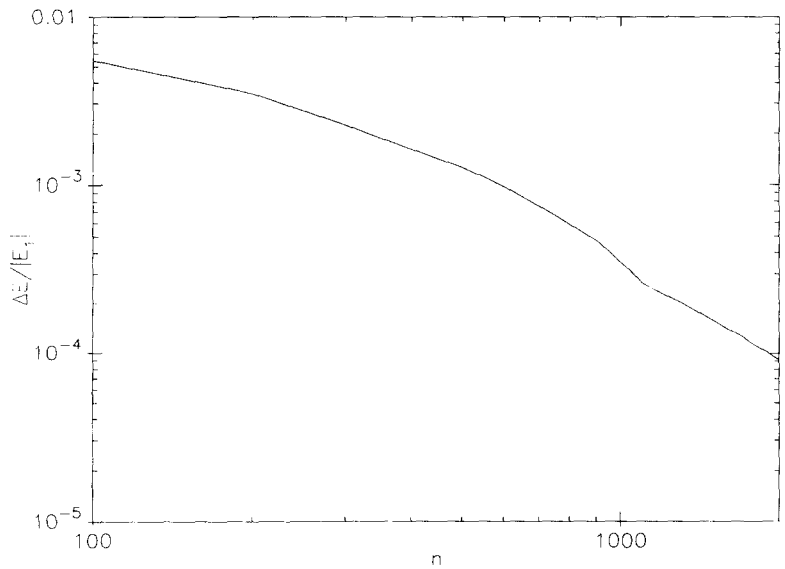

Fig. 9. Relative deviation $\Delta E/|E_1| = (E_1^{(n,m)} - E_1)/|E_1|$ as a function of the number of states n having B_2 symmetry. The model parameters are $L_x = L_y = 4$, $N_\uparrow = N_\downarrow = 4$ and $U/|t| = 4$.

Monte Carlo technique instead of Langevin dynamics to perform the importance sampling of Ising spin configurations. In the repulsive case $U > 0$ and for a non-half-filled band, QMC simulation methods severely suffer from minus-sign problems, especially at low temperature [16]. In particular, for band-fillings (excluding the half-filled band case) corresponding to open-shell situations, it

can be extremely hard to obtain reliable results for the ground-state properties. For the attractive model $U < 0$ the integrand is positive and accurate QMC results can be obtained [16].

For repulsive interaction $U > 0$, the minus-sign problem in PQMC simulation is most severe close to half-filling, even if a restriction to closed-shell systems is imposed. For reasons which are not clear to us, PQMC seems to give reproducible results for the energy. Due to the minus-sign problem, the statistical uncertainty on quantities other than the energy are often so large that no meaningful PQMC estimate can be made. In PQMC an estimator for a physical quantity takes the form [16] [see (2.3)]

$$\langle A\rangle_{\beta,m} \approx \langle\!\langle (\operatorname{sgn}\rho)\langle\psi_m|A|\psi'_m\rangle\langle\psi_m|\psi'_m\rangle^{-1}\rangle\!\rangle / \langle\!\langle (\operatorname{sgn}\rho)\rangle\!\rangle\,, \tag{5.11}$$

$$\langle\!\langle X\rangle\!\rangle = \sum_{\{\psi_i\}}{}' \sum_{\{\psi'_i\}}{}' X(\tau, m, \phi_0, \{\psi_i, \psi'_i\})|\rho| \Big/ \sum_{\{\psi_i\}}{}' \sum_{\{\psi'_i\}}{}' |\rho|\,, \tag{5.12}$$

where the prime on the summation symbols indicates that the sum runs not over all states but over the small subset generated by the simulation method. If the averaged sign, $\langle\!\langle \operatorname{sgn}\rho\rangle\!\rangle$, is not exactly one, all estimators (5.11) are biased and the amount of bias is unknown. Furthermore, the estimates of the numerator and denominator are not statistically independent since the same set of states is used to compute the averages. Experiments show that this correlation is essential to obtain meaningful results for the energy (in the case that there are minus-sign problems). In addition chapter 2 pointed out that due to the importance sampling, the energy computed by PQMC is neither an upper nor a lower bound. Thus, in view of all these difficulties, the accuraccy PQMC results should be taken with some caution.

The SD can be viewed as an importance sampling technique to optimize a trial wave function. Accordingly, the (approximate) ground-state energy should be the first physical quantity to look at. A collection of SD, exact and PQMC results for the ground-state energy are given in table 1. The number of important states N_R used by the SD method is a rather small fraction of the total number of states N (see corresponding entries in table 1). Apparently the SD technique is able to select the most important states out of the large number of states N. This suggests that the crucial assumption made in the introduction, namely that good results can be obtained with a (small) fraction of all states. may hold.

The data of table 1 further suggest that to reach the same level of accuracy, open-shell systems require more basis states than closed-shell cases. We have studied the effect of degeneracy in the non-interacting ($U = 0$) case and found that the SD technique has no problems dealing with highly degenerate matrices, a feature which it seems to share with the Jacobi method itself [1].

In chapter 3 it was shown that the SD method not only gives an approximation to the ground-state energy but also furnishes the necessary information to compute the (approximate) ground-state itself [see (3.13)]. The many-body ground-state wave function is constructed from the sequence of plane rotations and the expectation value of an operator X in the ground state is approximated by

$$\langle X\rangle^{(n,m)} = \sum_{i,j=1}^{n} \mathscr{U}_{1i}^{(n,m)} \mathscr{U}_{j1}^{(n,m)} \langle\phi_i|X|\phi_j\rangle\,. \tag{5.13}$$

We have used (5.13) to compute the momentum distribution and various correlation functions. Representative results are given in tables 2–4. There is excellent agreement between the SD and

Table 1
Comparison between ground-state energies (in units of $|t|$) as obtained from SD, from exact numerical diagonalization (exact) and from Projector Quantum Monte Carlo (PQMC) simulation. (L_x, L_y) are the numbers of lattice sites in the x- and y-directions and $(N_\uparrow, N_\downarrow)$ denote the numbers of up and down electrons, respectively. N_R is the number of important states gathered by SD and N the total number of states, not accounting for reductions due to rotational symmetry. Results obtained by exploiting the full symmetry of the model are marked by an asterisk (*).

(L_x, L_y)	U	$(N_\uparrow, N_\downarrow)$	Exact	PQMC	SD	N_R	N
3×3	4	(3, 4)	− 7.915		− 7.915	900	10584
3×3	20	(3, 4)	− 6.122		− 6.120		10584
3×3	4	(4, 5)	− 6.210		− 6.203		15876
3×3	8	(4, 5)	− 3.545		− 3.545		15876
4×4	4	(1, 13)	− 7.063		− 7.063		8960
4×4	4	(2, 2)	− 11.53		− 11.51	216	14400
4×4	4	(3, 3)	− 15.14		− 15.14	900	313600
4×4	4	(4, 4)	− 17.53	− 17.3	− 17.53	2×10^3*	3.3×10^6
4×4	4	(5, 5)	− 19.58[a]	− 19.6	− 19.58	7.3×10^3*	1.9×10^7
4×4	8	(5, 5)	− 17.51	− 17.5	− 17.40	4.5×10^3	1.9×10^7
4×4	−4	(5, 5)		− 32.6	− 32.59	9000	1.9×10^7
4×4	4	(6, 6)	− 17.73		− 17.70	9.7×10^3*	6.4×10^7
4×4	1	(7, 7)	− 21.39[b]	− 21.38	− 21.37	1.4×10^3*	1.3×10^8
4×4	4	(7, 7)	− 15.74[a]	− 15.7	− 15.45	7×10^4	1.3×10^8
4×4	4	(7, 7)	− 15.74[a]	− 15.7	− 15.49	4×10^4*	1.3×10^8
4×4	4	(8, 8)	− 13.62[a]	− 13.6	− 13.42	7×10^4	1.7×10^{13}
4×4	4	(8, 8)	− 13.62[a]	− 13.6	− 13.59	2.7×10^4*	1.7×10^{13}
4×4	−4	(8, 8)		− 45.4	− 45.35		1.7×10^{13}

[a] From ref. [23]. [b] From ref. [21]

Table 2
On-site pairing correlation function $C_{\text{Pairing}}(l) = L^{-1}\times\Sigma_i\langle c^+_{i,\uparrow}c^+_{i,\downarrow}c_{i+l,\downarrow}c_{i+l,\uparrow}\rangle$ of a 4×4 Hubbard model containing 5 electrons with spin up and 5 electrons with spin down, as obtained from SD and PQMC. The interaction $U = 4|t|$ and the number of states used by the SD is $N_R = 8\times10^3$. The projection parameter entering the PQMC method $\beta = 8|t|$ and the number of time slices $m = 64$. Statistical errors on the PQMC data have been estimated from 10 independent runs of 5×10^8 MC steps each.

$\|l\|$	SD	PQMC
0	0.04765	0.04794 (33)
1	0.01750	0.01772 (12)
$\sqrt{2}$	0.00153	0.00168 (5)
2	0.00153	0.00168 (5)
$\sqrt{5}$	− 0.00057	− 0.00058 (5)
$2\sqrt{2}$	0.01087	0.01062 (16)

PQMC data for the on-site pairing correlation function of a 4×4, system containing five electrons with spin up and five electrons with down, independently of the interaction strength ($U = 4|t|$ for table 2 and $U = 8|t|$ for table 3). Although minus signs are present, the PQMC simulation data is fairly accurate because the averaged sign is close to one. This is due to the fact that the system at

Table 3
On-site pairing correlation function $C_{\text{Pairing}}(l) = L^{-1} \times \Sigma_i \langle c^+_{i,\uparrow} c^+_{i,\downarrow} c_{i+l,\downarrow} c_{i+l,\uparrow} \rangle$ of a 4×4 Hubbard model containing 5 electrons with spin up and 5 electrons with spin down, as obtained from SD and PQMC. The interaction $U = 8|t|$ and the number of states used by the SD is $N_R = 6.5 \times 10^3$. The projection parameter entering the PQMC method $\beta = 8|t|$ and the number of time slices $m = 64$. Statistical errors on the PQMC data have been estimated from 10 independent runs of 5×10^8 MC steps each.

$\|l\|$	SD	PQMC
0	0.02436	0.02611 (96)
1	0.00924	0.01007 (45)
$\sqrt{2}$	0.00110	0.00133 (17)
2	0.00113	0.00132 (17)
$\sqrt{5}$	− 0.00058	− 0.00062 (15)
$2\sqrt{2}$	0.00370	0.00376 (36)

Table 4
The correlation function $C(l) = L^{-1} \Sigma_i \langle c^+_{i,\uparrow} c_{i,\uparrow} c^+_{i+l,\downarrow} c_{i+l,\downarrow} \rangle$ of a 4×4 Hubbard model containing 7 electrons with spin up and 7 electrons with spin down, as obtained from SD and exact diagonalization [21]. The interaction $U = |t|$ and the number of states used by the SD is $N_R = 2 \times 10^4$.

$\|l\|$	Exact	SD
0	0.1555	0.1531
1	0.2019	0.1946
$\sqrt{2}$	0.1958	0.2096
2	0.1739	0.1759
$\sqrt{5}$	0.1984	0.1910
$2\sqrt{2}$	0.1749	0.1768

hand has a closed shell. Correlation functions which, due to minus-sign problems, would be extremely hard to obtain by QMC are listed in table 4. Comparison of SD and exact diagonalization [21] indicates once more that SD is effective in collecting the important contribution to the ground state.

The SD results for systems up to 4×4 lattice sites convincingly show that SD can be used to find the important contribution to the ground state. The introduction already pointed out that a characteristic feature of this class of physical systems is that the number of states spanning the Hilbert space increases rapidly with the size of the system (assuming the density of particles to be fixed).

To explore the possibility of using SD to study systems that are not amenable to exact diagonalization, we have used SD for 6×6 and 8×8 lattices. Some SD and PQMC results for the energy are collected in table 5. Our current implementation of the SD algorithm for the Hubbard model is not yet as efficient as it could be. This is reflected in the maximum number of important states a run was allowed to use. The total number of states N and the number of important states N_R differ by many orders of magnitude. In view of this, and the intrinsic problems of PQMC, the agreement is satisfactory.

Table 5
Comparison between ground-state energies (in units of $|t|$) as obtained from SD method (taking into account the full symmetry of the square) and from Projector Quantum Monte Carlo (PQMC) simulation. N_R is the number of important states gathered by SD and N the total number of states.

(L_x, L_y)	U	$(N_\uparrow, N_\downarrow)$	PQMC	SD	N_R	N
6×6	4	(5, 5)		-29.99	392	1.4×10^{11}
6×6	4	(9, 9)		-41.45	2.2×10^{4}	3.6×10^{16}
6×6	4	(13, 13)		-40.77	2.3×10^{4}	5.3×10^{18}
8×8	4	(5, 5)	-34.3	-34.31	2828	5.8×10^{13}
8×8	4	(9, 9)	-54.6	-54.37	8789	7.6×10^{20}
8×8	4	(13, 13)	-66.8	-66.05	2.3×10^{4}	1.7×10^{26}
8×8	-4	(13, 13)	-91.1	-87.9	2.7×10^{4}	1.7×10^{26}
8×8	4	(25. 25)	-72.1	-67.00	2.3×10^{4}	1.6×10^{35}

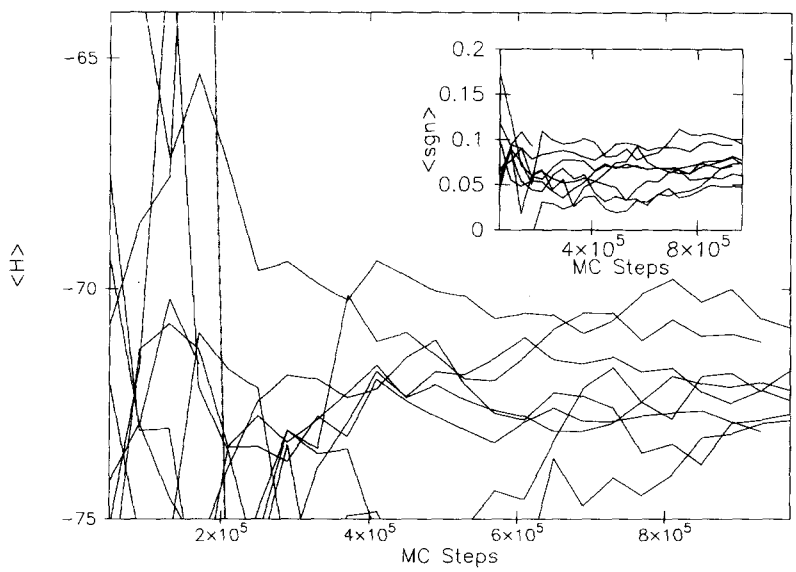

Fig. 10. PQMC results for the energy $\langle H\rangle_{\beta,m}$ and the average sign (inset) as a function of the number of Monte Carlo steps. PQMC parameters are $\beta = 4$ and $m = 32$. The model parameters are $L_x = L_y = 8$, $N_\uparrow = N_\downarrow = 25$ and $U/|t| = 4$.

Representative results for an 8×8 lattice are shown in fig. 10 where we present PQMC data for the energy (denoted by $\langle H\rangle$) and averaged sign (inset) as a function of the number of Monte Carlo steps. The averaged sign is of the order of 0.1 or less. Results for the energy, as obtained from statistically independent runs, show a trend to approach an average value of about $-72|t|$. The distribution of weights $d^{(n,m)}$ as obtained by SD is depicted in fig. 11. Detailed analysis reveals that states obtained by two particle–hole excitations dominantly contribute to the leftmost peak. The weights of four particle–hole excited states appear in the rightmost peak. As before, from the CPU-time dependence of this distribution, it is evident that in terms of importance sampling, the SD technique is performing as it should. Finally, fig. 12 shows the approximate ground-state energy $E^{(n,m)}$ as a function of the number of collected states n. For $n \leq 500$ the energy decreases rapidly. The SD algorithm is collecting the important states of the two-particle–hole class, of which there

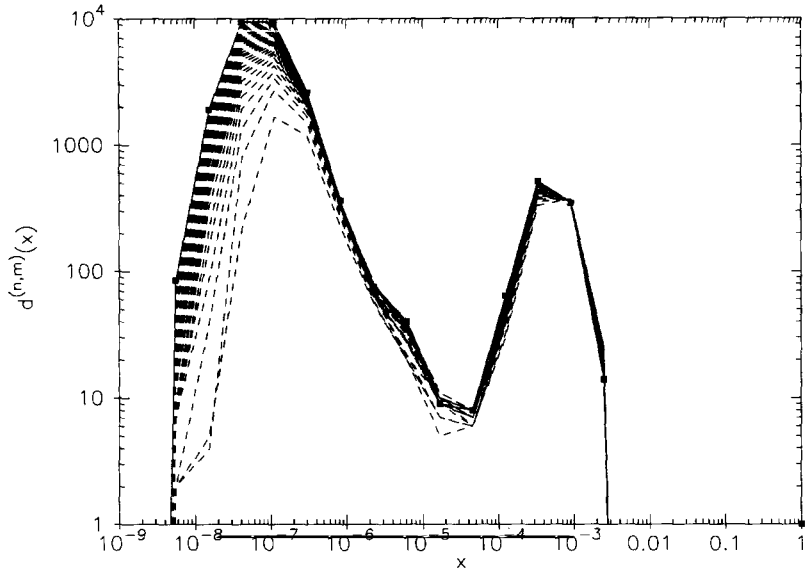

Fig. 11. Distribution $d^{(n,m)}(x)$ of the weights of the states ϕ_i as obtained from the approximate ground-state wave function $\Phi_1^{(n,m)}$ for increasing size of the set of states $S^{(n)}$. The solid line gives the result for $n = 2.7 \times 10^4$. The dashed lines represent distributions for various $n < 2.7 \times 10^4$, approaching the solid line as n increases. The model parameters are $L_x = L_y = 8$, $N_\uparrow = N_\downarrow = 25$ and $U/|t| = 4$.

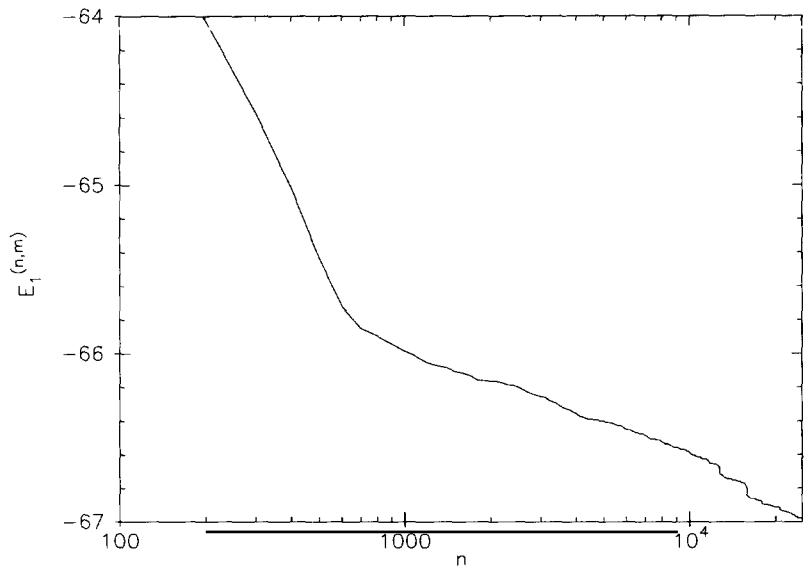

Fig. 12. The approximate energy $E_1^{(n,m)}$ as a function of the number of important states n. The model parameters are $L_x = L_y = 8$, $N_\uparrow = N_\downarrow = 25$ and $U/|t| = 4$.

are approximately 27 000. Then, in the second phase, it is scanning the vast space of states of the four-particle–hole type. The energy decrease is much slower but still substantial. It is clear that 20 000 states is not enough to approximate the ground state which, in view of the fact that there are more than 10^{35} states to choose from, should not come as a surprise.

6. Conclusions

A new algorithm has been presented to calculate the lowest eigenvalue and corresponding eigenvector of a real symmetric matrix. A rigorous proof of the correctness of the method has been given. Unlike conventional importance sampling techniques, the method discussed in this paper does not suffer from the minus-sign problem. The algorithm can exploit the spareness of the solution (if present) through the use of importance sampling. A mathematically well-defined criterion is used to guide the search for the most important contributions to the eigenvector corresponding to the lowest eigenvalue.

The stochastic diagonalization algorithm described in this paper is a novel method to minimize the Raleigh quotient $\langle\psi|H|\psi\rangle/\langle\psi|\psi\rangle$ with respect to the vector ψ. The standard approach to solve this minimization problem is to use conjugate gradients [22]. The conjugate gradient method requires storage for at least two vectors, i.e., ψ and $H\psi$. Even in the case that the number of "important" entries in ψ is relatively small, in general this feature is lost when H is applied to ψ. Therefore, in practice, conjugate gradient methods require storage of the order of the dimension of the Hilbert space. This is not the case for the stochastic diagonalization method for which storage scales with the number of important contributions to the ground state. Of course it is conceivable that for a certain representation the number of important contributions is of the same order of magnitude as the dimension of the Hilbert space. In that case there is little point in using stochastic diagonalization but, as explained in the introduction, other (stochastic) methods become useless as well if the Hilbert space becomes much larger than the available storage.

Ab-initio calculations for condensed matter and quantum chemistry models often lead to a generalized eigenvalue problem, resulting from the use of nonorthogonal basis states. Although the Jacobi method can be modified to deal with this situation [2], the question whether the stochastic diagonalization can be extended to solve more general eigenvalue problems is left for future research.

Acknowledgement

It is a pleasure to thank R. Broer, P. de Vries, E. Loh, K. Michielsen, W. Nieuwpoort, T. Schneider and W. von der Linden for simulating discussions. We are indebted to P. de Vries for critical readings of the manuscript and to A. Parola for providing us with exact diagonalization data. We are grateful to the IBM Research Laboratory, Zurich, for the hospitality extended to us during several summer visits. This work is partially supported by the Netherlands Organization for Scientific Research, FOM project 90.816 VS-G-C, and a supercomputer grant of the Netherlands Computer Facilities, NCF.

Appendix A. Background material

This appendix collects some results of matrix algebra essential to the development of the theory presented in chapter 3 and also supplies a proof of theorems 1 through 3. The theoretical justification of the importance sampling algorithm heavily relies on the use of the separation theorem [1, 2]. We simply state it here, using the notation introduced in chapter 3.

Separation theorem. The eigenvalues of $H^{(n)}$ separate those of $H^{(n+1)}$, that is

$$E_1^{(n+1)} \leq E_1^{(n)} \leq E_2^{(n+1)} \leq E_2^{(n)} \leq \cdots \leq E_n^{(n)} \leq E_{n+1}^{(n+1)} . \tag{A.1}$$

This, as well as the following result, is a direct consequence of the minimax characterization of the eigenvalues [1, 2].

Monotonicity theorem. If $C = A + B$, where A, B and C are real and symmetric $n \times n$ matrices having the eigenvalues a_i, b_i and c_i, respectively, arranged in nonincreasing order, then, for any i, j satisfying $1 \leq i + j - 1 \leq n$,

$$a_i + b_j \leq c_{i+j-1} , \qquad c_{n-i-j+2} \leq a_{n-i+1} + b_{n-j+1} , \qquad |c_i - a_i| \leq \|B\| , \tag{A.2a, b, c}$$

Theorem 1. The necessary and sufficient condition for $(e^{-\tau H})_{ij}$ to be positive for all $\tau > 0$ is $H_{ij} \leq 0$ for all $i \neq j$ [13].

Proof. First note that if $H_{ii} > 0$ for one or more i, adding a properly chosen constant ω will yield $H_{ij} + \omega\delta_{i,j} < 0$ for all i and j. Since $e^{-\tau H} = e^{-\tau(H+\omega)} e^{\tau\omega}$ and $e^{\tau\omega} > 0$ this does not affect the sign of $(e^{-\tau H})_{ij}$. Writing

$$e^{-\tau H} = \lim_{m \to \infty} (1 - \tau H/m)^m , \tag{A.3}$$

it is clear that the matrix product in (A.3) will contain positive numbers if $H_{ij} < 0$ for all i and j. The necessity of the above condition immediately follows from $(e^{-\tau H})_{ij} = \delta_{i,j} - \tau H_{ij}$ for $\tau \to 0$.

Theorem 2. If A is a positive-definite matrix with the property that $A_{ij} < 0$ for all $i \neq j$ then $A_{ij}^{-1} > 0$ [13].

Proof. Since A is positive-definite, the solution of $Ax = y$ is equivalent to minimizing [13]

$$F(x) = F(x_1, \ldots, x_n) = \sum_{i,j=1}^{n} x_i A_{ij} x_j - 2 \sum_{i=1}^{n} y_i x_i . \tag{A.4}$$

Assume that $y_i \geq 0$, for $i = 1, \ldots, n$ and consider $F(x)$ at some point x. Let $\mathcal{N}$ and $\mathcal{P}$ be the set of indices for which $x_i < 0$ and $x_i \geq 0$, respectively. Since $A_{ij} < 0$ and

$$F(x) = \sum_{i,j \in \mathcal{N}} x_i A_{ij} x_j + \sum_{i,j \in \mathcal{P}} x_i A_{ij} x_j - 2 \sum_{i \in \mathcal{P}} y_i x_i + 2 \sum_{i \in \mathcal{N}, j \in \mathcal{P}} x_i A_{ij} x_j - 2 \sum_{i \in \mathcal{N}} y_i x_i , \tag{A.5}$$

$F(x)$ can be reduced further by changing the sign of all the $x_j, j \in \mathcal{N}$. Thus at the minimum of $F(x)$, i.e. $x = A^{-1} y$, the $x_i, i = 1, \ldots, n$ can be taken to be non-negative. Next consider the solution of $Ax = y$ for any $y \neq 0$ satisfying $y_i \geq 0$, $i = 1, \ldots, n$. Rearranging the equation $Ax = y$ we have

$$A_{ii} x_i = y_i - \sum_{j \neq i}^{n} A_{ij} x_j , \quad i \neq k . \tag{A.6}$$

At the minimum of $F(x)$ all the x_i are non-negative. By hypothesis all off-diagonal elements of A are negative and there exists at least one $y_k > 0$. Therefore for $i = k$, (A.6) implies that $x_k > 0$ as all the $A_{ii} > 0$ because A is positive-definite. However, since $x_j > 0$ for $j = k$, (A.6) shows that all the x_i are strictly positive. Thus, for all possible choices of $y \neq 0$ satisfying $y_i \geq 0$, $i = 1, \ldots, n$, the solution of $Ax = y$, i.e. $x = A^{-1}y$, has all strictly positive elements (all $x_i > 0$). This then implies that A^{-1} must have all strictly positive elements, for if $(A^{-1})_{ij} \leq 0$ for some pair (i,j), take $y^{\mathrm{T}} = (0, \ldots, 0, \tilde{y}_j, 0, \ldots, 0)$ with $\tilde{y}_j \geq 0$ and compute $x_i = \Sigma_{k=1}^{n} (A^{-1})_{ik} y_k = (A^{-1})_{ij} \tilde{y}_j \leq 0$ to conclude that there is a contradiction with the previous result that all $x_i > 0$.

Theorem 3. Let λ_1 be the smallest eigenvalue of the real and symmetric matrix $A = \begin{pmatrix} x & y^{\mathrm{T}} \\ y & Z \end{pmatrix}$. Then the smallest eigenvalue $\mu_1 = \mu_1(t)$ of the matrix $B(t)$ of the matrix $B(t) = Z - (x - t)^{-1} yy^{\mathrm{T}}$, $x \neq t$ satisfies

$$\lambda_1 + \frac{\lambda_1 - t}{(x - t)(x - \lambda_1)} \| y^{\mathrm{T}} v_1(\mu_1) \|^2 \leq \mu_1 \leq \lambda_1 + \frac{\lambda_1 - t}{(x - t)(x - \lambda_1)} \| y^{\mathrm{T}} v_1(\lambda_1) \|^2 , \tag{A.7}$$

$$B(t)v_1(t) = \mu_1(t)v_1(t), \quad v_1^{\mathrm{T}}(t)v_1(t) = 1 .$$

Proof. Writing $Au = \lambda u$ as

$$\begin{pmatrix} x - \lambda & y^{\mathrm{T}} \\ y & Z - \lambda \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = 0 , \tag{A.8}$$

and eliminating the variable u, the secular equation $\det(A - \lambda) = 0$ can be written as

$$\det(B(\lambda) - \lambda) = \det[Z - (x - \lambda)^{-1} yy^{\mathrm{T}} - \lambda] = 0 , \quad x \neq \lambda . \tag{A.9}$$

The eigenvalues λ_i of A are the solution of (A.9). Solving

$$\det [B(t) - \lambda] = 0 , \quad x \neq t , \tag{A.10}$$

yields the eigenvalues $\mu_i(t)$ of $B(t)$. Note that $\mu_1(\lambda_1) = \lambda_1$. The minimax or, in this case, variational principle applied to $B(t)$ gives

$$\min_{\|v\| = 1} v^{\mathrm{T}} B(t) v = \mu_1 , \tag{A.11a}$$

and, by construction,

$$\min_{\|v\| = 1} v^{\mathrm{T}} B(\lambda_1) v = \lambda_1 . \tag{A.11b}$$

Since

$$B(t) = B(\lambda_1) + [(\lambda_1 - t)/(x - t)(x - \lambda_1)] yy^{\mathrm{T}} , \tag{A.12}$$

$$f(x_0) + g(x_0) = \min_x [f(x) + g(x)] \geq \min_x f(x) + g(x_0) \geq \min_x f(x) + \min_x g(x) , \tag{A.13}$$

where x_0 is the value of x minimizing $f(x) + g(x)$, minimizing (A.12) and invoking the first inequality in (A.13) gives the lower bound on μ_1. Interchanging t and λ_1 in (A.12) and minimizing yields the upper bound on μ_1.

Appendix B. Counterexamples

The conditions under which the modified Jacobi scheme isolates the smallest eigenvalue are highly nontrivial. In principle, the combination of the modified Jacobi scheme and matrix inflation leads to the smallest eigenvalue if with each inflation step, the off-diagonal matrix element $\alpha_1^{(n)}$ does not vanish [see eq. (3.18)]. The examples below illustrate what may go wrong if this is not the case.

Let us start by considering the 2×2 matrix

$$A^{(2)} = \begin{pmatrix} x & y \\ y & x \end{pmatrix}, \tag{B.1}$$

where without loosing generality, we may assume that $y > 0$. The smallest eigenvalue of (B.1) is $x - y$. Now we inflate the matrix by adding one row and column, yielding for example

$$A^{(3)} = \begin{pmatrix} x & y & z \\ y & x & z \\ z & z & x \end{pmatrix}. \tag{B.2}$$

Application of the plane rotation

$$U = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{B.3}$$

yields

$$U^{\mathrm{T}} A^{(3)} U = \begin{pmatrix} x - y & 0 & 0 \\ 0 & x + y & \sqrt{2}z \\ 0 & \sqrt{2}z & x \end{pmatrix}. \tag{B.4}$$

The eigenvalues of (B.2) are $x - y$, $x + \frac{1}{2}y \pm (2z^2 + \frac{1}{4}y^2)^{1/2}$. The matrix elements of (B.2) are such that the plane rotation annihilates the (1, 2) and (2, 1) elements and "by accident" also clears the (1, 3) and (3, 1) elements. In the notation of chapter 3, $\alpha_1^{(3)} = 0$. Now there are two possibilities. If $|y| < |z|$, then $x - y > x + \frac{1}{2}y - (2z^2 + \frac{1}{4}y^2)^{1/2}$, i.e., $x - y$ is *not* the smallest eigenvalue of (B.2), and the method has failed. However, if $|y| > |z|$ then $x - y$ is the smallest eigenvalue and up to this point the method seems sound.

The above example may suggest that by annihilating the element $(1, j)$ of maximum modulus, the modified Jacobi algorithm may still yield the smallest eigenvalue. As the following example demonstrates, this is not always the case. Consider the matrix

$$A^{(4)} = \begin{pmatrix} x & y & z & -z \\ y & x & z & -z \\ z & z & x & u \\ -z & -z & u & x \end{pmatrix}, \tag{B.5}$$

and to simplify the discussion, assume that $y > 0$, $z > 0$, and $u > 0$. We wish to show that even though we took all precautions (which in this case means taking $y > z$, $y > u$) not to run into the same problems as with matrix (B.2), it is nevertheless possible that the modified Jacobi method fails to isolate the smallest eigenvalue on the (1, 1) position.

Application of the plane rotation

$$U = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \\ -1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{B.6}$$

yields

$$U^{\mathrm{T}} A^{(4)} U = \begin{pmatrix} x-y & 0 & 0 & 0 \\ 0 & x+y & \sqrt{2}z & -\sqrt{2}z \\ 0 & \sqrt{2}z & x & u \\ 0 & -\sqrt{2}z & u & x \end{pmatrix}. \tag{B.7}$$

The off-diagonal elements in the fourth row and column have been choosen such that a plane rotation involving elements (3, 3), (4, 4), (3, 4), (4, 3) annihilating the latter two elements also clears the elements (2, 4) and (4, 2). Indeed, the plane rotation

$$V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 0 & -1/\sqrt{2} & 1/\sqrt{2} \end{pmatrix} \tag{B.8}$$

transforms matrix (B.7) in

$$V^{\mathrm{T}} U^{\mathrm{T}} A^{(4)} U V = \begin{pmatrix} x-y & 0 & 0 & 0 \\ 0 & x+y & 2z & 0 \\ 0 & 2z & x-u & 0 \\ 0 & 0 & 0 & x+u \end{pmatrix}. \tag{B.9}$$

The eigenvalues of (B.9) are

$$\lambda_1 = x - y\,, \qquad \lambda_2 = \lambda_1 + \tfrac{1}{2}(3y - u) - \sqrt{[\tfrac{1}{2}(y+u)]^2 + 4z^2}\,, \tag{B.10a, b}$$

$$\lambda_3 = \lambda_1 + \tfrac{1}{2}(3y - u) + \sqrt{[\tfrac{1}{2}(y+u)]^2 + 4z^2}\,, \qquad \lambda_4 = x + u\,. \tag{B.10c, d}$$

By construction $\lambda_1 < \lambda_3$ and $\lambda_1 < \lambda_4$ but $\lambda_1 < \lambda_2$ if and only if $1 - u/y > 2(z/y)^2$, a condition which in general need not be satisfied. We have therefore shown that there exists an example for which the modified Jacobi method, combined with the matrix inflation technique, will not yield the smallest eigenvalue if, during the inflation process, $\alpha_1^{(n)} = 0$, for all $n = n_0 + 1, \ldots, N$ with $1 < n_0 < N$.

References

[1] J.H. Wilkinson, The Algebraic Eigenvalue Problem (Clarendon, Oxford, 1965).
[2] B.N. Parlett, The Symmetric Eigenvalue Problem (Prentice-Hall, Englewood Cliffs, NJ, 1981).
[3] J.K. Cullum and R.A. Willoughby, Lanczos Algorithms for Large Symmetric Eigenvalue Computations (Birkäuser, Boston 1985).
[4] E.R. Davidson, J. Comput. Phys. 17 (1975) 87.
[5] J. Olsen, P. Jørgensen and J. Simons, Clem. Phys. Lett. 169 (1990) 463.
[6] J.M. Hammersley and D.C. Handscomb, Monte Carlo Methods (Methuen, London, 1964).
[7] K. Binder, in: Monte Carlo Methods in Statistical Physics, ed. K. Binder (Springer, Berlin, 1979).
[8] K. Binder and D. Staufer, Applications of the Monte Carlo Methods in Stastical Physics, vol. 36, Topics in Current Physics, ed. K. Binder, (Springer, Berlin, 1984).
[9] M. Suzuki, in: Quantum Monte Carlo Methods, ed. M. Suzuki (Springer, Berlin, 1986).
[10] H. De Raedt and W. von der Linden, Inter. J. Mod. Phys. C 3 (1972) 97; Phys. Rev. B 45 (1992) 8787.
[11] H. De Raedt and A. Lagendijk, Phys. Rep. 127 (1985) 233.
[12] E.Y. Loh, Jr. and J.E. Gubernatis, Electrons Phase Transitions, eds W. Hanke and Y.V. Kopaev (Elsevier, New York, 1990).
[13] R. Bellman, Introduction to Matrix Analysis (Maple Press, York, PA, 1960).
[14] K.E. Schmidt and M.H. Kalos, in: Applications of the Monte Carlo Methods in Statistical Physics, ed. K. Binder (Springer, Berlin, 1984).
[15] J.H. Hetherington, Phys. Rev. A 30 (1984) 2713.
[16] H. De Raedt and W. von der Linden, Monte Carlo Methods in Condensed Matter Physics, ed. K. Binder (Springer, Berlin, 1992).
[17] P.W. Anderson, Science 234 (1987) 1196.
[18] W. Jones and N.H. Marsh, Theoretical Solid State Physics (Dover Publ., New York) p. 985.
[19] M. Hamermesh, Group Theory and its Application to Physical Problems (Addison-Wesley, Reading, MA, 1962).
[20] S. Sorella, S. Baroni, R. Car and M. Parrinello, Europhys. Lett. 8 (1989) 663.
[21] A. Parola, Private Communication.
[22] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, Numerical Recipes (Cambridge Univ. Press, New York, 1986).
[23] A. Parola, S. Sorella, M. Parrinello and E. Tosatti, Physica C 162–164 (1989) 771.